Diploma Work

OE-NIK	Student’s name:	**Jlassi, Mohamed**
2022	Student’s registration number:	**T/007242/FI12904/N**

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbuda University, Budapest
(Hungary)
JOHN VON NEUMANN
FACULTY OF INFORMATICS

Diploma Work

Data acquisition and analysis from sensor-based dental braces using smartphones. Adatgyűjtés és elemzés érzékelő alapú fogszabályozóból okostelefonok használatával

OE-NIK Student's name: **Jlassi, Mohamed**
2022 Student's registration number: **T/007242/FI12904/N**

Óbuda University
John von Neumann Faculty of Informatics
Institute of Biomatics and Applied Artificial Intelligence

DIPLOMA THESIS
TOPIC DESCRIPTION

Student name: **Mohamed Jlassi**
Registration number: T/007242/FI12904/N
Neptun's code: [redacted]

Title of diploma thesis:

Data acquisition and analysis from sensor-based dental braces using smartphones
Adatgyűjtés és elemzés érzékelő alapú fogszabályozóból okostelefonok használatával

Internal supervisor: Prof. Dr. Miklós Kozlovszky
External supervisor

Deadline: 15th of May, 2023

Topic description:

Since its foundation, smartphones provided a huge number of opportunities turned into solutions that helped organizations and individuals in matters of communication, entertainment, social media, or health monitoring. The student should design and develop a remote patient monitoring mobile application, which is able to collect usage parameters data received from a proprietary sensor embedded in the dental braces.
The smartphone should be able to exchange wirelessly the data with the sensor, and the received data should be analyzed and visualized to support the dentist with useful information for the optimal treatment. The collected data should be stored locally and additionally it should be forwarded to the central data collector server.

The diploma thesis should cover:

- literature review of the medical background of the dental braces, why we need to monitor them and technics available in the market used before to monitor,
- requirement specification of the given problem,
- design and implementation,
- test and validate the results,
- analyze the results.

ÓBUDAI EGYETEM * Neumann János Informatikai Kar * 1. Budapest

P.H.

.....
Dr. György Eigner
head of institute

Term of limitation: **15th of May, 2025.**

The diploma thesis is adequate and can be submitted:

.....
external supervisor

.....
internal supervisor

STUDENT'S DECLARATION

I the undersigned student hereby declare that this thesis is the result of my own work; I have disclosed the references and tools used in an identifiable manner. The results indicated in my degree project / thesis completed may be used by the university and the institution announcing the task for their own purposes free of charge.

Dated: Budapest, May, 5th 2022

.....

student's signature

Log of consultations

Name of the student:	Neptun code:	Program:
Jlassi Mohamed	[redacted]	MSc. Computer Engineering

Phone number: Address:

[redacted]

Title of the thesis work:

Data acquisition and analysis from sensor-based dental braces using smartphones

Internal supervisor: External supervisor:

Dr. habil Miklós Kozlovszky .....

Please use BLOCK CAPITAL LETTERS to fill in this form!

Ocassion	Date	Topics discussed	Signature
1.	12/14/2020	OVERVIEW ABOUT THESIS WORK	
2.	01/14/2021	LITERATURE REVIEW	
3.	03/16/2021	DISCUSSING SYSTEM REQUIREMENTS & SPECIFICATION	
4.	07/05/2021	REFINING THE SYSTEM REQUIREMENTS & SPECIFICATION AND VALIDATING THE SYSTEM DESIGN	

This log should be signed on each consultation (by a supervisor), 4 times altogether.

The student fulfilled the requirements of the Thesis work subject, I allow his/her participation on the presentation/defence[1].

.....

Internal supervisor

Budapest, 07/05/2021

[1] Underline the appropriate one.

Log of consultations

Name of the student:	Neptun code:	Program:
Jlassi Mohamed	[redacted]	MSc. Computer Engineering

Phone number: Address:

[redacted]

Title of the thesis work:

Data acquisition and analysis from sensor-based dental braces using smartphones

Internal supervisor: External supervisor:

Dr. habil Miklós Kozlovszky .....

Please use BLOCK CAPITAL LETTERS to fill in this form!

Ocassion	Date	Topics discussed	Signature
1.	09/22/2021	FINALISING THE SYSTEM DESIGN	
2.	10/29/2021	FINALISING THE SYSTEM IMPLEMENTATION AND DISCUSSING THE TESTING PART	
3.	11/26/2021	PRESENTING THE OVERALL THESIS-PROJECT	
4.	12/04/2022	DISCUSSION THE THESIS REPORT AND THE PRESENTATION	

This log should be signed on each consultation (by a supervisor), 4 times altogether.

The student fulfilled the requirements of the Thesis work subject, I allow his/her participation on the presentation/defence[1].

.....

Internal supervisor

Budapest, 12/04/2022

[1] Underline the appropriate one.

Table of Contents

List of abbreviations

BLE	Bluetooth Low Energy
PH	Potential of Hydrogen
GPS	Global Positioning System
SDK	Software Development Kit
UI	User Interface
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
CSS	Cascading Style Sheets
HTTPS	Hypertext Transfer Protocol Secure
MVC	Model–view–controller
IT	Information Technology
UUID	Universally Unique Identifier
REST	REpresentational State Transfer
API	Application Programming Interface
RPC	Remote Procedure Call
MAC	Media Access Control
OS	Operating System
IDE	Integrated Development Environment
AWS	Amazon Web Services
N/A	Not Applicable

Abstract

The number of people with teeth alignment problems is very high and always increasing because these problems don’t concern only kids but also adults. From these problems, we can mention crooked teeth, overlapping, overcrowded teeth, the size difference between the top jaws and bottom ones known as Malocclusion, etc. Sensor-based devices usage has significantly grown in the market and especially in relation to the medical field with the goal of monitoring, analyzing, and providing the medical staff with useful information to decide on the treatment needed for patients. This kind of technology saves time for both doctors and patients and prevent unexpected situations from happening and most of all easy access to the needed information without the need for physical contact, especially for the braces which can be tricky and time-consuming. This thesis work goal is to develop mobile software that collects wireless data concerning the usage parameters of the braces from a proprietary sensor embedded in the braces, analyze them, and present them. The collected data is stored both locally and remotely on the central server.

Keywords: teeth alignment problems, sensor base devices, wireless data collection, remote monitoring

Introduction

Since their foundation, smartphones have been a game changer in the world of technologies because it provided the users practically with mini personal computer in their pocket which can do anything in a matter of seconds thanks to the sensors, cameras and all the technologies built inside. Also, the mobile platform gained and still gaining more interest among the other platforms because nowadays it's very rare to meet someone who doesn't possess at least one smartphone or more. In addition to its worldwide spread, the smartphones offer all kind of applications from different categories that supports the daily activities of users at a personal level or even at work.

As the demand grows for smartphones as the development of new applications grows everyday giving the users a wide range of applications that helps users in matters of communication, social media, entertainment and also body health monitoring thanks to wearable devices like smartwatches or other devices which contains mainly sensors to collect the user's data in a certain context: movement, heartbeats, steps done every day, etc. These devices are capable to establish the connection to smartphones wirelessly via Bluetooth generally which enable the application to collect the data, process it and display it in the application developed for that need.

The field of applications dedicated to healthcare and that support healthcare personnel especially the doctors with their monitoring, analyses and decision making for their patients, have known a real growth the last decade and that impacted the healthcare system positively thanks to the evolution of applications and also the technological evolution of smartphones capabilities and taking advantage of devices containing embedded sensors which can even be present in human body, in that way the capture of data by these sensors would be very precise and continuous which gives a huge amount of reliable and concrete data that can be after that accessed wirelessly and easily from the smartphone without the need of any contact or the unbind of the device from the body. This data would be processed and presented in a mobile application in a simplified way.

The growth of demand of similar applications generally with sensor-based devices is resumed in the gain of time and of the personnel which can be exploited in other more useful tasks and in addition to that prevent unexpected situations from happening through the continuous monitoring. Therefore, thanks to this kind of applications, doctors wouldn't be needed to have a consultation daily, just in a single consultation through a mobile application the doctor can access the device and see the log and the analysis of the patient during any period of time he needs just by fixing the dates from and until when in the application.

Also, as the demand of less contact and social distancing grows lately due to the Covid-19 pandemic spread, this kind of application can be very helpful and the solution in avoiding useless contact and keeping both the patients and the healthcare personnel safe from the spread of the virus.

In this work, I'm going to develop a mobile application dedicated both for the doctors and for the patient that aims to collect data from proprietary sensor embedded in dental braces wirelessly, process them and present them in the application interface on a smartphone mainly in the goal of helping the doctor monitor and evaluate the improvement of the patient brace treatment.

I have adopted the following plan to write my thesis work:

- The first chapter is an introductory chapter dedicated to a general presentation of the topic, description of the existing systems, the technical choices, the methodology and the formalism adopted during the implementation process.
- The second chapter will feature the requirement specification and the analysis of the project, as well as the different use cases of the system.
- The third chapter is devoted to the design specification, where I describe the architecture of the application and all the diagrams related to the application.
- The fourth chapter will imply the realization of the application, the environment: languages, tools, and Framework used in the implementation phase and the description of the system with the illustration of some interfaces of the product developed.
- The fifth chapter will be the system testing includes the environment and data testing to get the desired result also some screenshots will show this result.
- At the end eventually, the conclusion would finally summarize the work carried out and describe any possible prospects for the project.

1. Preliminary study

1. Purpose

In this chapter, I will begin with a description of the problem. Then, I will present a short description of dental braces usage and why we need them generally. On the other hand, I will explain the goal of monitoring dental braces, describe the advantages of this monitoring, and the problems which can be monitored through the braces. In addition to that, the sensor-based devices and their role in the modern medical field will be put as the subject of a literature study alongside the existing systems.

1.1. Summary of the problematic

Since old ages, humans have been developing means to treat teeth misalignments which can occur due to multiple facts hereditary, bad habits, etc. The most commonly used treatment through decades now is the appliance of dental braces. The main weaknesses of this treatment are that it can take years in some cases and so it can be time and effort-consuming for both the doctor and the patient because of the multiple checking-in sessions to monitor the progress of the treatment physically. Also, It lacks some important information like the daily usage of the braces, the temperatures registered during the usage, etc.

To resolve these problems, I proposed a system that is divided mainly into two components. Firstly, a dental brace embedded inside of it has a sensor capable of recording data useful for the dentists to monitor the patient's braces treatment progress. The data is related basically to the temperature measured within the braces through time and the sensor is able to communicate this data wirelessly through Bluetooth Low Energy. Secondly, a mobile application that can retrieve the data through Bluetooth, store it permanently, and show visualizations based on it. This solution can provide the doctor with more insights and so help him evaluate thoroughly the treatment progress.

1.2. Literature review

Due to their strong onboard computational capacity, large memory, wide displays, and open operating systems that promote application development, the current versions of smartphones are increasingly seen as portable computers rather than phones. Smartphones are increasingly allowing normal access to knowledge in previously unimaginable ways, including in the field of medical education.[1] Once upon a time, phones were only used to communicate with others via phone calls and text messaging. They offer several pre-loaded software packages. However, the rise of smartphones has opened up new possibilities for using mobile technology in regular clinical practice.

Smartphones are a useful tool for fast reference or when accessing a desktop computer is not possible due to its mobility, ability to connect to the internet everywhere, speed, multiple features, and simplicity. As a result, cellphones may be used as rapid reference tools for daily life tasks such as train ticket buying, as well as daily spending tracking, checking bank balances, etc. Despite the broad availability of smartphones and applications for general purposes, certain apps are expressly intended

for the medical industry. Medical apps have been one of the fastest expanding genres of applications and include a variety of orthodontic-specific apps.

The growth in demand for smartphone usage in the medical field was aligned with the growth in demand for healthcare wearable tech which got so high recently since the most essential advantage is that they provide users with the data they need to better control their health outcomes. With the help of these devices, the software that came with them, and as well the use of wireless connectivity by healthcare providers we were able to gain a better understanding of the health status, allowing us to make better decisions. The use of the sensor allows us to track progress toward our health objectives by monitoring the different health-related data sent by these devices' sensors. Due to these previously mentioned facts, the wearable device market is continuously expanding, which according to some estimates will be worth around $12 billion in 2021.[2]

Thrope and Shannon created the first wearable device in 1966. This wearable device was a small computer that measured the speed of a wheel and then transmitted the results to an earpiece. As a result, humans were able to develop computers that were lighter and smaller than ever before. Many firms and researchers, including Pulsar, have worked with Steve Mann to explore using technology to improve human life.[3]

Since wearable devices gather a large quantity of data, which implies that big, personalized data is becoming more real, it is important for providers to think about and discover ways to collect the data properly, protect the data, and use it more effectively. Measuring consumed calories, blood pressure, heart rate, and sleeping hours are some of the well-known basic applications that wearable devices may provide in the healthcare area. Initially, there was reluctance in the healthcare industry to modify and accept new concepts, but the development of wearable devices and their usefulness to healthcare has demonstrated the contrary currently. According to Lewy, the usage of wearable devices in healthcare can provide extra information by integrating data from many sources, supplementing existing clinical data on the EHR, and generating knowledge.[4]

With the rise in smartphone and wearable devices use, the medical industry has embraced the technology, with a variety of applications already accessible to patients to help them quit smoking and manage pain, to name a few examples. Applications for doctors in orthopedic surgery, ophthalmology, and radiology, as well as apps for medical students, have previously been published.[1]

With this in mind, it is not unexpected that these portable devices are being employed in an increasing number of fields in the health industry. One example of a wearable device is a contact lens that checks and analyzes blood sugar levels in diabetics. Another example is a Bluetooth thermometer, which transmits the patient's temperature to a mobile phone.

In the dentistry field, wearables are also a possibility. Temperature, PH values, and other factors, for example, might all be measured. Who knows, maybe these small

practical components can allow dentists to improve patient loyalty in the future through better diagnosis. In any case, recent research indicates that people are more interested in sharing their health data with doctors and even health insurance firms via smartphones and other devices[5].

These software applications provide new avenues for doctors to use technology in their clinical practice, as well as for patients to gather information regarding their treatment. As a result, the patient is always involved in their treatment path, not only during treatment choice. All of these possibilities are referred to as "telemedicine" or "eHealth" in the medical profession, and "teledentistry" in the dental field. The concept of "teledentistry" is no longer a distant reality: recent surveys indicate that 78 percent of patients will begin using teledentistry within the next five years. Teledentistry can be defined as the use of new technologies such as videos, images, and communication technologies such as video and voice calls[6]. In addition, it assists in the delivery of dental care, diagnosis, consultation, treatment, and patient education. It can involve virtual consultations and remote patient monitoring, making treatment less expensive and more efficient for both patients and doctors. According to Petya et al., 89 % strongly believe that interactive dental care may enhance their oral health, which can be easily provided with several dental applications.[7]

In fact, orthodontists are eager to incorporate new findings and technology into their clinics these days. Because of the COVID-19 constraints, a substantial number of patients may be seeking orthodontic treatment with no physical contact appointments while allowing the orthodontist to evaluate their treatment progress.[8]

There are still numerous barriers to overcome before this technology becomes prevalent; for example, how to secure private health information from being hacked or leaked, as well as how to create proper rules for its use are just a few examples of the issues this technology will face. With the advancement and growth of wearable devices, one issue that has arisen, particularly for small organizations, is what to do with all of the data collected from the wearable devices. Some firms can manage and analyze massive volumes of data, such as Google and Amazon, and in the future, small enterprises may rely on the procedures or even the infrastructure developed by data-handling giants.

1.3. Dental braces usage

The social nature of human existence has required the quest for ways and means of restoring, substituting, and arranging front teeth on an aesthetic basis since ancient times. Today, as in the past, a great deal of emphasis has been placed on looks, particularly the attractive smile. Our smile is our business card, and it is one of the aspects that determine our self-esteem and ability to interact with others. Since the dawn of humanity, ancient civilizations such as the Egyptians, the ancient Romans, and Greeks all of them utilized primitive types of braces to straighten crooked teeth in life or to preserve straight teeth in death and the afterlife Gold, silver, and/or catgut were common materials for braces in ancient times. Platinum, silver, steel, wood, and ivory have all been utilized as brace materials throughout history[9]. Modern braces were

firstly invented in 1819 by Christophe-Francois Delabarre, a Frenchman who revolutionized the discipline of dentistry with important breakthroughs. The invention of the predecessor to braces as we know them today was thanks to Delabarre. He created a woven wire that was worn for a period of time across both the upper and the lower teeth to progressively straighten them.[2] It won't be until 1997, that Zia Chishti came up with an invisible transparent plastic retainer that can accomplish the same straightening action as standard metal braces without the use of archwire or periodic adjustment, inspired by his own orthodontic treatment. He was also the source of the Invisalign invention, alongside his fellow Standford graduate Keley Wirth, which was a treatment consisting of clear hospital-grade plastic fashioned to suit the teeth and adjusted every couple of weeks to influence the desired movement. This invention was made available beginning in 2000.

Orthodontic therapy is an investment in the health of both our teeth and our bodies. The potential of orthodontic procedures in the treatment of dentofacial and maxillofacial abnormalities in children, adolescents, and adults is continually growing due to the use of variously fastened orthodontic equipment on the oral carriage. A braces therapy is never too late. This treatment may take a longer time for adults since their bones won’t be growing any longer. On the other hand, it would be easier at younger age lower than 18 years old[10].

1.4. Common Teeth Problems Treated through Dental Braces

In this section, I will describe some of the most recurring problems related to teeth misalignment mainly treated through the appliance of dental braces and which eventually can be monitored.

1.4.1. Tooth Crowding

Dental crowding is described as a mismatch between the size of the jaw and the size of the teeth, resulting in dental imbrication and rotation.

1.4.2. Diastema

Diastema refers to a space between your teeth. Any of an individual's teeth might be affected. It's particularly visible when there's a gap between your top front teeth because of its position. It can cause phonetic issues in some circumstances, especially when the gaps are large.

1.4.3. Overbite

An overbite, also known as buck teeth, is a dental misalignment. When your upper front teeth protrude (pop out) past your lower front teeth, it's called a gummy smile. The causes leading to overbite can be hereditary which means that it is inherited through the family. Also, teeth's alignment might be affected by genetic factors such as jaw shape.

1.4.4. Underbite

The jaw is divided into two parts: upper and lower. When the bottom section of your jaw protrudes further than the top part, you have an underbite. It can be inconvenient, creating issues with chewing, digestion, and other ailments.

1.4.5. Crossbite

A crossbite is a kind of misalignment of the teeth, in which the upper teeth are positioned inside the lower teeth. This misalignment can impact a single tooth or a group of teeth, and it also can impact either the front teeth or the back teeth.

1.4.6. How recording the temperature and movement can help in teeth problems monitoring

So, monitoring the movement of the teeth during the treatment of this disease can provide the dentist with the needed daily progress of teeth movement. Along, with movement measurement, temperature measurement can be useful to indicate the compliance of the patient in the disease treatment by detecting whether the brace is worn which means there's an increase in the temperature or just the ambient temperature which means it's not worn[11]. Also, the temperature records can give insights to the doctor into the interaction of the mouth during the treatment and detect any kind of issues like inflammation that can occur.

1.5. Why do we need to monitor the braces?

As we saw in the previous section the teeth problems that occur all around the world nowadays on daily basis and the treatment used commonly is through braces which are developing as we have seen previously. Nonetheless, this development didn't cover entirely the monitoring part because some dentists are still depending on old traditional methods to follow up on the brace's treatment of their patient's progress. Each orthodontic treatment lasts between one and three years. Different types of orthodontic methods exist, depending on the patient's issue and the age at which treatment begins. Orthodontic appointments are typically scheduled every three weeks to a month and a half.[8] So, this raised the need for another solution to help monitor the orthodontic treatments and this is where new technologies intervene. The parallel development of mobile applications and other technologies gave us the possibility to monitor the health and treatment conducted on every part of the human body through wearable devices such as sensors, smartwatches, etc. The main advantages behind using such devices and more precisely having a sensor embedded in the braces are:

- Gain of time: This affects both the doctor and the patients which means the doctor wouldn't have to do very frequent appointments with the patient to monitor. Also, monitoring the braces would help to detect the unexpected problems, the needed fixes, and the setups to be done at an early stage of the treatment. As a result, these advantages can help the dentist achieve the goals set with the patient in a more reduced time than the traditional treatments.

- Gain of efforts with more efficiency: The strict follow-up on patients' cases and the progress done can be effort consuming and sometimes isn't very precise due to invalid information extracted through the physical consultations or provided by the patient about his teeth hygiene. Therefore, using a mobile application to extract the needed data related to his specific treatment and statistics about the patient's usage of the braces without the need even for a physical contact can

reduce the efforts dedicated to this task and on the other hand get more precise data and insights of the teeth treatment progress.

- Complete documentation and records of the treatment: Monitoring the braces would help doctors keep the full record of patients from the beginning of the treatment until the end which can help even with digitalizing the way the data is persisted and it can be very useful for advanced studies and making improvements in future cases. Because nowadays data is a very valuable asset and the more data, we have the more the analysis and insights behind it can be revealed.

- The reduction of physical contact: Due to the emergence of new viruses all around the globe citing the global pandemic virus Covid-19, which is transferred through human contact, the need for less contact during treatment and consultations is growing. Therefore, this kind of monitoring would be the solution because it allows dentists to avoid unnecessary contact with the patients while still getting the needed data easily and efficiently.

1.6. Existing systems

Sensing technology in dentistry has advanced. Clinical applications in dentistry, such as diagnosis and monitoring diseases and health in oral tissues, are made easier by sophisticated imaging and sensing technology. These recent advances in optical sources and detectors have novel applications such as non-destructive testing, high-speed and low-cost electronic circuits, and novel signal processing methods. Among these technologies, sensor-based braces which can enable wireless monitoring surfaced in the market with multiple solutions.

To improve orthodontic performance, several sensors have been designed to track patient compliance, dental cleanliness, load force on teeth, and tooth movement.

1.6.1. INVISALIGN

Invisalign is a set of transparent plastic aligners that are worn over the teeth. The aligners should be worn every day, for 22 hours and only removed for eating, drinking, brushing, or flossing. Yes, unlike braces, Invisalign clear aligners are removable you must remove them to level up from set to set as your teeth gradually shift into their optimal positions. That is how Invisalign corrects bite disorders including overbites, underbites, and crossbites, resulting in wonderfully straight and healthy teeth.

Invisalign was linked to a smartphone mobile application afterward called **Dental Monitoring**. So, Dental Monitoring is a mobile app that allows dentists to remotely follow the Invisalign treatment. Every week, the patient will use his smartphone to take a snapshot scan of his smile on the dental monitoring app — it's similar to taking a selfie, but there’s a full instruction on the manufacturer's website. The photo is submitted in the application, where it is reviewed and compared to the treatment plan set by the dentist. If the dentist is satisfied with the progress and fit of the patient’s

Invisalign clear aligners, he may continue without needing to return to the clinic. The patient has also the possibility to send messages to his dentist directly from the app if he has any issues or believes his aligners aren't fitting properly. Figure 1[12]

Figure 1 - Dental Monitoring Mobile App

1.6.2. TheraMon

A variety of attempts have been made to produce precise and reliable devices for measuring the time patients wear their orthodontic equipment, including headgear and other detachable appliances. TheraMon® microsensor (Handelsagentur Gschladt, Hargelsberg, Austria or Forestadent, Pforzheim, Germany), which has been installed in many detachable appliances such as dental braces, is the most recent gadget developed to objectively monitor adherence to appliance wear guidelines. Every 15 minutes, the microsensor reports the temperature. At each appointment, the orthodontist may access the data from the microsensor and utilize a dedicated software application (desktop application) to interpret the results based on wear-time assumptions. The program now 'validates' wear-time as temperature values ranging from 33.5 to 39°C[13]. Figure 2[14]

Figure 2 - TheraMon Monitoring App

1.6.3. Grin Remote Monitoring Platform

Grin is a complete digital orthodontic platform that allows orthodontists to remotely monitor their patients' dental health. It comprises a smartphone app called Grin App and an intraoral adaptor that retracts the cheeks to allow a complete view of the patient's mouth called Grin Scope. Patients are able to record teeth and jaw movements during video capture if given the proper instructions. Following that, orthodontists can evaluate the recorded intraoral video using the online doctor portal.[15]

This orthodontic platform can be classified basically the starting point for doctors to determine the appropriate teeth alignment treatment and may also be used in the follow-up process when combined with the Invisalign plastic aligners which makes it very similar to the Dental Monitoring application mentioned previously.

1.6.4. Comparative study

In Table 1, I will describe each of the existing systems in the market based on the existing and non-existing features of these solutions. Then, it would be followed up by a solution that will be proposed for this thesis project and what is the expected features compared to the other existing solutions.

Solution	Existing Features	Missing Features
INVISALIGN + Dental Monitoring	• Plastic aligners removable + Mobile application • Assistance through the usage of the aligners. • Remote follows up through: • Teeth Scanning • Messaging between the doctor and the patient	○ No embedded sensors in the aligners to provide data such as the temperature and movement ○ No helpful data apart from the teeth pictures ○ It Depends on the doctor to check out the teeth scan shots and produce conclusions.
TheraMon	• Removable braces containing a sensor. • A dedicated sensor reading station to transfer data from the sensor to the computer. • desktop application. • Persistence of data for every patient • Visualization of data like braces wearing times in form of charts • Persistence of basic data related to the sensors and patients and doctors • Useful for early detection of inefficient treatment	○ The application is a desktop application so it's not portable and less practical ○ The connectivity to the application requires a reading station which enlarges the system components list. ○ Application available only for the doctors
Grin Remote Monitoring Platform	• Mobile application + intraoral adaptor • The ability to add and monitor the Invisalign plastic aligners treatment • Remote follows up through:	○ Same as the Dental Monitoring app in case of Invisalign usage means no embedded sensors in the aligners to provide data such as the temperature and movement ○ No helpful data apart from

	• Teeth Scanning • Messaging between the doctor and the patient	the teeth pictures and videos ○ It Depends on the doctor to check out the teeth scan shots and produce conclusions.
BraceCheck(This thesis Application)	• Removable braces containing a sensor. • Mobile application. • Bluetooth connectivity between the sensor and the application • Persistence of data for every patient • Visualization of recorded temperature data: old data and Real-time data through charts and tables. • Persistence of basic data related to the sensors and patients and doctors • Not only the doctor can access the application, but also the patient can since he has some functionalities like reporting usage problems. • Statistics	○ No movement sensor ○ No offline availability

Table 1 - Comparative study between the existing solutions and the envisaged solution

2. Technical choices

2.1. Overview of Mobile application types

In this section, we are going to present the two known types of mobile applications which are the native and the hybrid, and show the differences between both and the choice adopted.

2.1.1. Native

A native mobile application is an application dedicated to smartphones coded in a programming language appropriate to the operating system, such as Swift for iOS and Java or Kotlin for the Android platform. Natively developed mobile apps offer a high level of performance and reliability that's why most mobile games are developed in the native platform.

This sort of program, on the other hand, is costly to produce since it is bound to a single operating system, forcing the developers to generate from scratch duplicate copies that run on other platforms.

- **Android:** Android is a mobile operating system and a software development platform for smartphones in addition to tablets and other mobile devices. It can operate on a wide range of devices produced by multiple manufacturers. Working with Android requires its software development kit (SDK) that enables developers to create android applications by coding original code and assembling software components. Android also has the advantage over IOS in terms of customization because its system is based on the Linux Kernel which makes it open to modifications and customizable for different phones.

- **IOS**: It is commonly related to iPhones but it's not the operating system of the iPhones only. IOS is an operating system created by Apple and it is dedicated to iPhones, iPads, and iPods. The iOS SDK allows developers to create apps for the iPhone and iPad. For Mac users, the SDK is a free download; however, it is not accessible for Microsoft Windows PCs.

  The SDK provides developers with sets that provide them access to various functions and services provided by iOS devices, such as software characteristics and hardware. To test apps, get technical support, and publish applications through the App Store, developers must enroll in the Apple Developer Program.

  Xcode is the IDE used by iOS developers to code iOS applications in the supported languages such as Objective C or Swift.

2.1.2. Hybrid Mobile Applications

Hybrid mobile apps are similar to regular native apps both installed on the same device. What sets them apart is that they combine aspects from native apps, which are software designed for a specific platform such as iOS or Android, with components from web apps, which are webpages or websites that function like apps accessible via a browser on the Internet but not installed on the device.

In other terms, it’s the encapsulation of web content on the mobile user interface. It means there are web programming languages used like JavaScript alongside html5 code, also SDKs like Apache Cordova which the mostly used and known.

Hybrid applications are easier to develop but this ease may cause a less performant and less stable application compared to native applications and hence less suited to each platform.

A hybrid application has the advantage of being easier and faster to create than a native app. Because there is just one version to evaluate for many platforms, maintaining the app will be simpler. However, all these benefits come at a cost: the application's performance and stability suffer as a result of the system's lack of adaptability to each platform. LinkedIn, Gmail, Twitter, and Instagram are some well-known examples of hybrid apps, and they are High-performance apps used worldwide every day.

Another major benefit of hybrid application development is that you only have to manage one version regardless of how many platforms you need to deploy. Therefore, if you wish to add new features or upgrade the application, you will only have to do it once. These added changes will be applied on each platform whereas in a native application. It takes longer and is more expensive.

Limitations of hybrid apps:

The most known limitation noticed across hybrid apps is the fact that the user experience will not be as good as it would be if native platforms were used. It needs to make compromises because it is being built for two OSs at the same time.

Moreover, it’s not able to exploit the device's native features. Plug-ins must be implemented to exploit more efficiently the capabilities of the Android and iOS systems. These plug-ins must be codded in the operating system's programming language. This increases consequently the difficulty and the complexity of the app development project.

While using a hybrid app, there is a noticeable lack of smoothness. As an explanation, the content displayed on the app is originally retrieved from the company's website. So, the majority of the data is generally kept on a server, and as a result, Latency may occur.

Other examples of limitations regarding the hybrid apps, there is:

- **Hard to keep up to date**: Given the complexity of the app's architecture, it's tough to keep it up to date.
- **Less security**: The degree of security isn't as high as it would be with a native program.
- **Offline mode not available:** Unlike native apps, offline mode isn’t possible for hybrid apps which are web apps basically and so they need to be connected to the internet all the time to work.
- **Dependency on 3rd party tools**: Hybrid development requires the use of third-party tools (Cordova, Flutter).

From the previously cited advantages and disadvantages, we can conclude that the hybrid app may be released faster than native software. It conforms to the constraints of business imperatives' often tight timelines. It's monetizable and has ratings in the App Store and Google Play Store. A compelling argument to develop a hybrid app.

2.2. Firebase Framework

2.2.1. What’s Firebase?

Firebase is a backend-as-a-service (BaaS) that provides a wide range of features and components for improved mobile and web application development[16]. Because of the amount of usability it provides, many business owners and developers favor Firebase. In other terms, it is a platform that allows the rapid development of web and mobile applications.

The purpose of the creation of Firebase.google.com in 2011 by James Tamplin and Andrew Lee is to save professionals and individuals from engaging in the complex process of creating and maintaining a server architecture. This service removes the need for users to create APIs and manage servers. So, developers may customize Firebase to meet specific requirements.

In Firebase, you'll find intuitive APIs bundled into a single SDK. These APIs, in addition to saving you time, allow you to reduce the number of integrations you need to manage through your application.

- **Firebase Databases**

Firebase provides two databases. The initial Firebase database product is the Real-Time Database, while Cloud Firestore is an updated version of the Real-Time Database.

- **Firebase's real-time database** allows developers to quickly store and sync data in real-time. It also enables users to access the database even if they're not connected to the internet. Data is stored as JSON on Firebase, and it is also synced between clients.
- **Firebase's Cloud Firestore** is a NoSQL cloud database that may be used to store and synchronize data for both server-side and client-side development purposes. Cloud Firestore makes it possible to construct mobile, web, and server applications flexibly. It's also useful for syncing data amongst real-time apps. Google Cloud and Firebase are also connected with Firestore.

In addition to the database features mentioned before, Firebase offers to the user other features which are the following:

- **Firebase Authentication:**

This tool provides easy-to-use SDKs, back-end services, or user interface libraries. These libraries allow you to authenticate your users[17].

Typically, manually configuring an authentication system takes several months. Thereafter, a team must be hired for maintenance. With Firebase, things are different. Setting up the system only takes a few hours, even if it takes care of delicate operations like merging accounts. Several methods are available to you to authenticate your users, including the use of Email/password, Google account, Facebook, Twitter, phone number, etc.

- **Firebase Cloud Functions**

Firebase Cloud Functions is a serverless platform that allows developers to run backend code that responds to Firebase and HTTPS request components. Cloud Functions are in charge of securing user logic and integrating the Firebase platform. This is a crucial feature[18].

- **Firebase Cloud Storage**

Firebase Cloud Storage is an object storage service with a wide range of features for app development. It's a cost-effective service that provides Google-level security for file downloads and uploads. Users can use cloud storage to save user-generated content and media files[19].

2.2.2. Backend technologies comparison study

To be able to choose the right database for this thesis project, I made a table of comparisons between the different databases used in the development market nowadays compared to the Cloud Firestore, which is I will illustrate in Table 2:

Providers	Differences
SQLite / Firestore	• Firestore is compatible with both Android and iOS / SQLite is only for Android. • SQLite database is a basic file that is saved locally on the device. Firebase may store data on a server to make it available to everyone and also provides local storage as SQLite does.
MySQL / Firestore	• Firestore is an auto-scaling database / MySQL is RDBMS. • When it comes to managing large data sets, Firebase is a pleasing option since NoSQL horizontally scales data and is significantly quicker than MySQL.
DynamoDB / FireStore	• Firestore is a real-time database / DynamoDB that works in JSON. • Firestore saves data in Google cloud / DynamoDB saves data in Amazon Cloud. • When it comes to storing new data, Cloud Firestore's data model is a little more flexible, whereas DynamoDB needs the creation of tables and primary keys beforehand.
MongoDB / Firestore	• MongoDB is open-source / Firestore is not open-source. • MongoDB is more popular for big data and high-performance use cases, whereas Firebase is more used for smaller apps.

Table 2 - Backend Technologies Comparison study

2.3. MVC architecture

The development of the user interface for my mobile application is based on the MVC architecture. MVC is the abbreviation of Model-View-Controller. Today, it is very widely used in the web development industry because it helps build maintainable and scalable web applications.

As its name says this architecture divides the software application in general into three components: the Model, the view, and the Controller. I'm going to simplify how this architecture will be applied to our project and how its three components interact with each other.

- The Model:

The model contains the data manipulated by the program. It handles the data and ensures its integrity. It also offers methods to update this data (insertion, deletion, update, search). It also offers methods to retrieve the data and do business operations which would be triggered through the application interface elements existing in the View component.

- The View:

The view establishes the interface for the user. It contains elements like text boxes, buttons, etc. The first task for the View is to display the data coming from the model. Its second task is to receive all user actions (mouse click, input selection, buttons, …). Its various events are sent to the controller. The view can also offer the ability to the user to change the view.

- The Controller:

The controller is responsible for the synchronization of the model and the view. It is the receiving point of all user events and based on the event it triggers the action to be performed. If an action requires a data change, the controller requests the data change from the model and then notifies the view that the data has changed so that it can update. Some user events do not concern the data but the view. In this case, the controller asks the view to modify itself. It always follows the same algorithm:

1. Receive the user's request (for example, a click of a button)
2. Analyze the consistency of the request according to the current state of the application.
3. Trigger the requested model business action or the action that manages cases of an invalid concatenation if the request does not need to be.
4. Retrieve the result of the executed action.
5. Build and present the response view to the user.

This architecture approach is very useful and saves time during application maintenance and evolution. Furthermore, it enables flexibility to organize the application's development inside a single team so multiple tasks can be developed by different developers on the different layers.

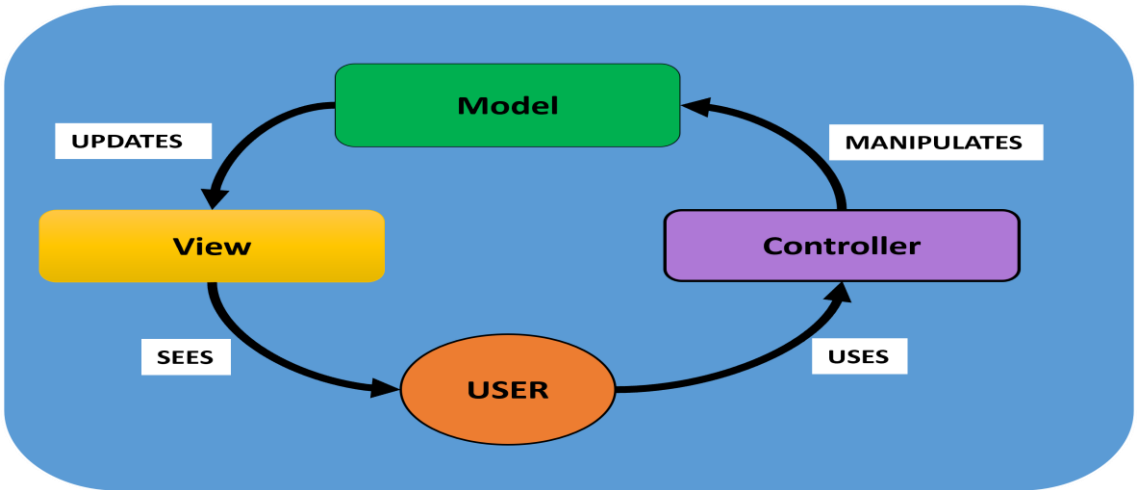

Figure 3 - MVC Architecture

2.4. Ionic Framework

Ionic Framework is a collection of tools and technologies that allow you to rapidly create hybrid mobile applications. It uses AngularJS for the web application side of the framework and Cordova for the native application building side.

So, to be able to develop a mobile application using Ionic Framework, the needed knowledge would be Web development tools such as Angular, HTML, CSS, etc.

Consequently, cross-platform programs are neither totally native nor entirely web-based. Although the program has access to native APIs, the design is made up of web views.

To build mobile applications from web content, ionic uses Apache Cordova to build on top. Because Ionics's capacity to improve user interface and user experience, as well as the most crucial component, simplifying mobile app development, is mainly thanks to Apache Cordova. In addition, the improved touch gestures, side menus, list views, and tabs are available for developers to use in just a few simple steps.

Angular is a well-known framework that makes it simple to create responsive web apps. And now, Ionic extends the potential of Angularjs to the app dev. Ionic allows us to develop mobile apps using the Angularjs framework by offering a comprehensive library of components as well as the tools required to convert Angular TypeScript scripts into a collection of JavaScript, HTML, and CSS, which Apache Cordova can then assemble into a native mobile app.

This open-source framework enables the creation of applications that may be deployed in a variety of environments, such as websites or mobile apps for platforms such as Android or iOS.

Since it allows many various plugins for development, Ionic allows us to use plugins to build an ideal mobile application. Moreover, Cordova is characterized by the hundreds of plugins that it offers as add-ons that developers can integrate into ionic projects, which allows accessing native device APIs such as the camera, the GPS, etc. It can be used to improve the features and functionality of your application.

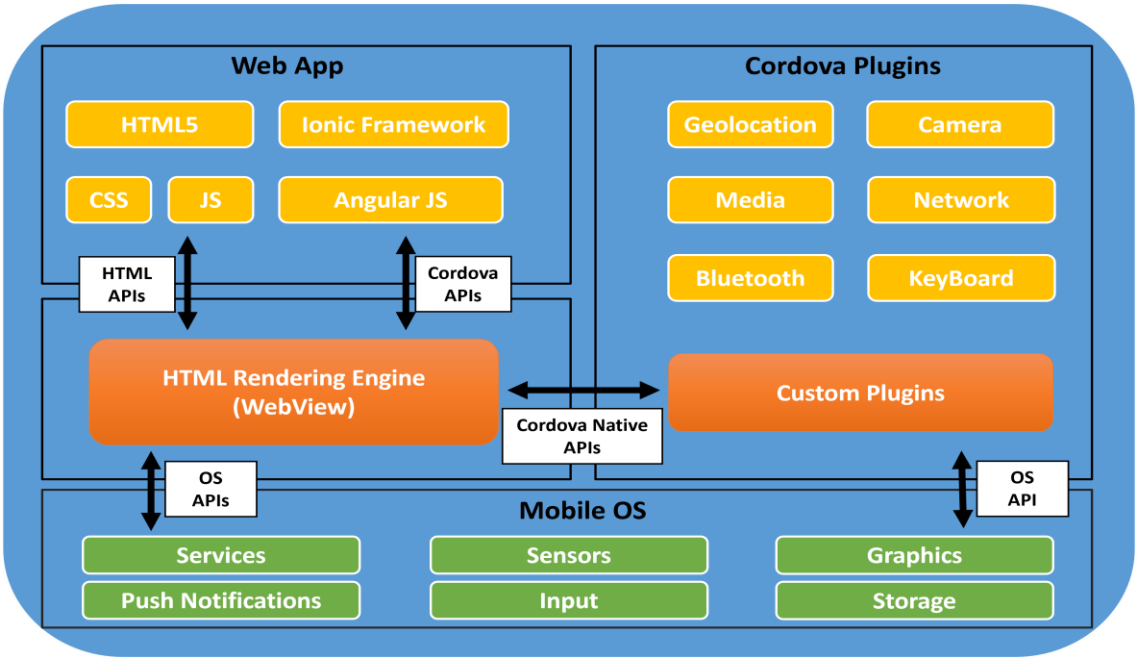

Figure 4 - Ionic application architecture

2.4.1. Apache Cordova

Apache Cordova is a collection of mobile APIs that allow an application's developer to use JavaScript to access native device capabilities like the camera and the vibrator. The application developer may use Cordova to create an app without having to write native code (Java, Objective-C, etc.). Instead, web technologies are employed in the application, which is hosted locally. Cordova may be used on a variety of platforms, including iOS.

2.4.2. Cordova Pugin

Plugins are an essential component of the Cordova ecosystem. They offer a communication interface for Cordova and native components, as well as connections to standard device APIs. This allows you to run native code from within JavaScript.

Apache Cordova project provides a collection of plugins called the Core Plugins. These core plugins allow your program to access device features like the battery, camera, contacts, etc.

To be noted too, there are no plugins included when you start a Cordova project. This is now the default setting. Any plugins you want to use, including the basic ones, must be added explicitly.

To construct the hybrid application in my project, we first need a component that lets us interact with the host device's native part, which in my case is the Bluetooth component, to connect to our braces-embedded sensor and handle all interactions with the braces using native code. Finally, the Cordova plugin must perform the following functions:

- Connect to the braces and herby connect to the sensor itself.
- Read the needed data from the memory of the sensor.
- Send back data to the mobile application using Cordova APIs.

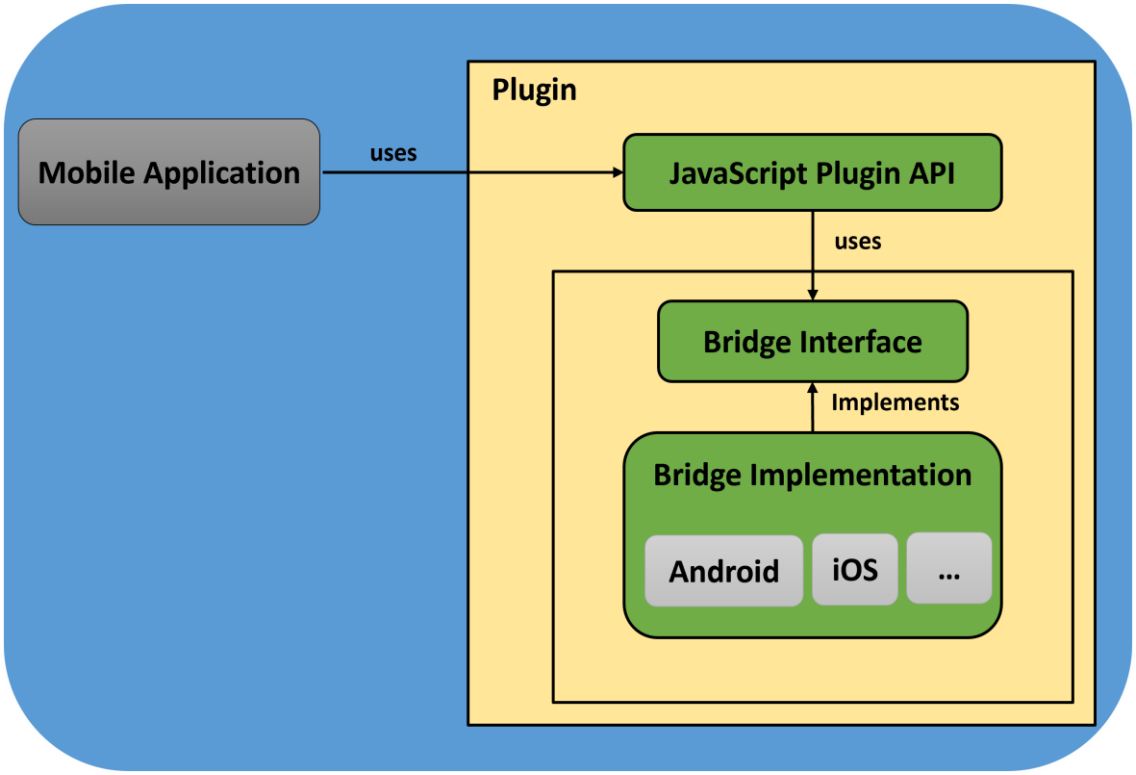

Figure 5 - Cordova Plugin Architecture

2.4.3. Web App

A web application is an application that is deployed on a server and accessible using a web browser. Users do not, however, need to install the software, unlike traditional software. As if it were a website, the program is accessed by a web browser. However, technical advancements have allowed the Progressive Web App to arise (PWA). A Progressive Web App is a web app with integrated functionality via "Service Worker," which was formerly reserved for native apps.

The advantages of a PWA are numerous. On the one hand, because programming languages are more prevalent, it offers potentially reduced development costs than a native or hybrid application (HTML 5, JAVA, PHP, REACT, Angular ...). Its development is based on API segmenting and associating the building blocks of an application. This design allows you to combine the greatest features of each technology used in the business architecture to create the ideal split between a native/mobile app

and a website. And from the other hand, because it can be accessed through a browser, it has the advantage of being cross-platform, meaning it can be accessed from any device: desktop, tablet, or mobile. Through Cordova, PWAs may access native capabilities like offline usage, GPS on-demand, geolocation, Bluetooth, NFC, push notifications, cache management, camera, and microphone.

2.4.4. Bridge Interface

The Bridge interface establishes a connection between the Cordova application's webview and the native platform on which it is operating. It specifies the native code requirements. A JavaScript file containing the input point that calls the action function on the native side of the service class is what's used to show the bridge interface. After successfully completing the execution on the native side, the application must get the results through the bridge interface using Cordova API SuccessCallBack, and in the opposite situation, Cordova API FailureCallBack must be triggered.

2.5. Bluetooth Low Energy

Bluetooth Low Energy in a nutshell is a technology that can be integrated into products because it consumes so little power that it can be built around a small, long-lasting battery. BLE or also known as Smart Bluetooth was presented by SIG in June 2010. BLE came as a continuation of the traditional Bluetooth, but with slightly different usage purposes. BLE dramatically optimizes communication over short distances between devices that do not involve large data transfers, and it is an effective technology for monitoring and controlling applications where the amount of data to be exchanged is often minimal[20].

Modern devices such as phones or computers are all equipped with Bluetooth and Wi-Fi which gives the possibility to devices to connect to a central hub which is generally a router then to the internet or other devices connected to the same hub. In Table 3, I will give a short comparison between Bluetooth and Wi-Fi.

	Bluetooth	Wi-Fi
Frequency	2.4 GHz	2.4 GHz, 3.6 GHz, 5 GHz
Range	60 m and 10 m indoor (Bluetooth 4.0)	It differs, the 2.4GHz can reach 46 m (indoor).
Data rate	25 Mbps (Bluetooth 4.0)	54 Mbps -1.3 Gbps
Communication	Direct between devices	The communication between two devices or more requires their connection to a common wireless router or an access point.
User support	The maximum number of devices which be connected to one Bluetooth is 7 devices	Theoretically, for most wireless routers/access points 255 devices.

Table 3 – Comparison between Bluetooth and Wi-Fi

2.5.1. Proposed solution process

Before diving into the technical aspect of the proposed solution and for a better understanding later of the system requirements and what should the system have as an outcome, I will describe the process on which eventually the application will be based.

In fact, a patient who is going to need the braces for his treatment will be given the braces which have the embedded sensor inside them. After the treatment starts, the doctor will eventually have multiple appointments with the patient. During an appointment, the usual way without our solution is the physical check done by the doctor and the notes of the appointment will be saved in the patient's file. Whereas in the proposed solution, the doctor will have the mobile application installed on his smartphone. He will need just to take out the braces from the patient and then he can connect to the sensor residing in the braces through the application. Once the phone and sensor are connected, the doctor will be able to visualize the incoming data from the sensor and, he will have the possibility to visualize old data stored data. This process is portrayed in Figure 7.

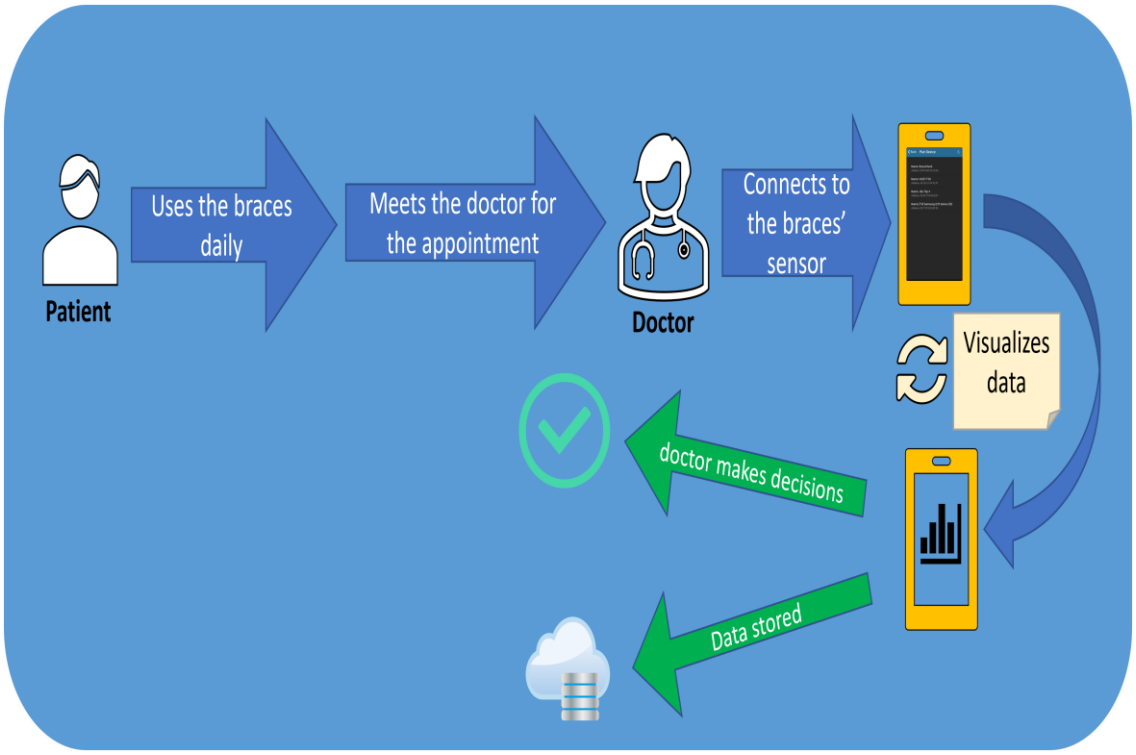

Figure 6 – Proposed Solution Process

2.5.2. Proposed solution

In order to solve the problem mentioned before through the technologies that were chosen for this project, the proposed solution consists mainly of providing a possibility for the sensor embedded in the braces to communicate the data with the mobile application through Bluetooth which will be treated than by the application to produce charts and eventually store the data permanently as we saw I the previous section. Figure 8 describes the proposed solution for establishing the communication between the sensor which is the transmitter and the smartphone which is the receiver.

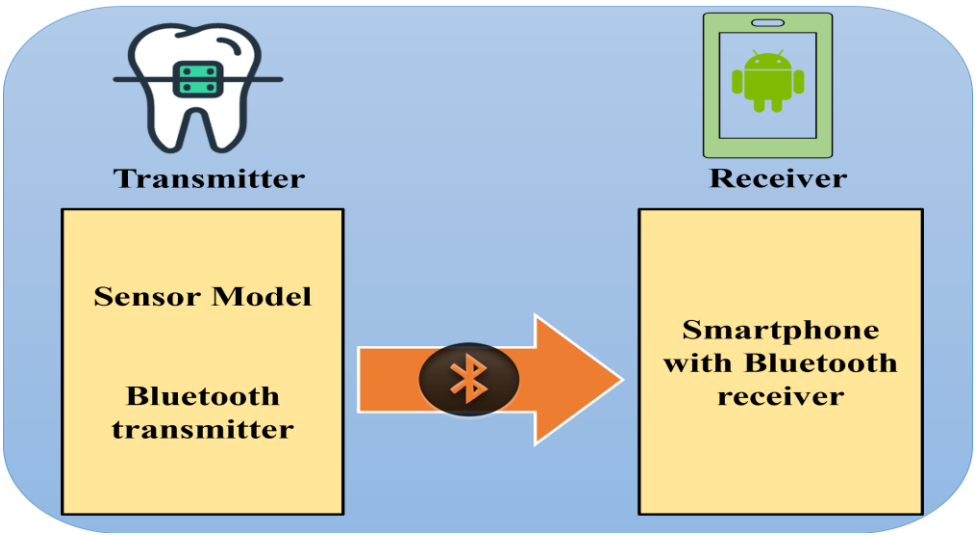

Figure 7 - Proposed solution Block Diagram

3. System Requirements and specification:

3.1. Purpose

In this chapter, I will describe and specify the study of the needs which is a crucial phase that allows us to define the application's different requirements. So before starting the realization of a project and guaranteeing its success, it is first necessary to carry out a well-defined needs study to specify functional and non-functional requirements, actors, and case diagrams of use.

3.2. Actors' identification

An actor is the presentation of a role played by an external person or process that interacts with a system. In this application, we can identify the following actors who will interact with the platform:

Doctor: the biggest part of the application is dedicated to the doctor. He can add a patient, connect to the dental braces of the patient, synchronize, and visualize patient data, also he can update his profile information.

Patient: After being authenticated the patient is able to view his profile and update it. In addition, the patient will be able to report if have any problems with the dental braces and request an appointment.

Administrator: this actor will assume the responsibility of managing the data related to the system overall through the Firebase application dashboard accessible through a certain URL. This dashboard provides all the needed tools to perform the task of administrating and managing the system data.

Both the patient and the doctor are inheriting from the user because similar attributes with some differences. Also, in terms of interaction between the two actors, every actor has his own user interfaces which are since they have different functionalities. But the only interaction which can happen is when there is a new brace related to a patient us paired, the doctor is responsible to specify that that certain brace is related to the patient.

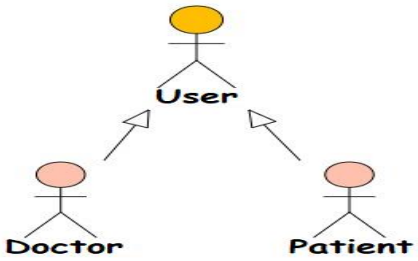

Figure 8 – Types of users

3.3. Functional Requirements

The functional requirements are the requirements that respond to well-defined points of the specification of the app. They are expressing an action that should affect the system in response to a request. A requirement that specifies an action that the system must be able to perform, regardless of physical constraints; a requirement that specifies the

behavior of the system in input and output. It is a requirement from the standpoint of the user.

A functional need consists in defining what the system will achieve in terms of the Business Layer. It leads to the development of use case diagrams. In this context, we present the functional needs of this project:

1. Authentication: This functionality will be responsible to authenticate the user to the application which is the most crucial part in terms of the application security and ensures only authorized and verified users connect to the application. The authentication will be simple and can be done in the classic way Email/Password or through the user's social media accounts like Facebook, Twitter or through his mailing accounts like Gmail. After being authenticated, depending on the user type he will be redirected to the appropriate page whether he's a doctor or a patient.
2. User Management: As in the case of the majority of applications through the different existing platforms there's a need for the module of user management which includes the creation, the update, and the delete of a user.
3. Data Management:
    3.1. Load data from the dental braces: This functionality is done by establishing the connection with the braces and then retrieving the data gathered by the sensor which are basically motion and temperature data related to the user in addition to other data like the battery level percentage of the sensor.
    3.2. Load stored data from the database: This functionality ensures the extraction of data from the database that will be after that presented to the user in multiple forms.
    3.3. Save Data: This functionality ensures the persistence of the received data in the database which basically happens when the device is connected to the mobile application and starts to send the data.
    3.4. Visualize data: This functionality will be responsible to visualize data in form of charts or tables that highlight the data received from the sensor embedded in the dental braces.
4. Add Sensor to patient: This functionality is done by the doctor in case of a new braces' sensor, the doctor should be able to affect the sensor to the patient during the process of connecting to it.
5. Profile Management:
    5.1. Profile consultation: This functionality enables the user to view his profile information.
    5.2. Profile update: This functionality enables the user to update his profile information after viewing it.

6. Statistics Visualization:

   In most modern applications, statistics became an essential part to develop as it's the case in our application. This functionality will provide for this version of the project just the number of readings of data done through the patient's device for the day selected.

7. Appointment suggestion: This functionality offers to the patient the possibility to suggest a date for meeting the doctor in case of an urgent situation or any other extraordinary cases.

8. Problem reporting: This functionality offers to the patient the possibility to report problems related to the brace's usage.

3.4. Non-functional requirements:

These are the requirements that define the system such as performance, maintenance, extensibility, and security constraints. Among these needs we mention the following:

- ❖ Security Constraint: This application must ensure:
  - The confidentiality of information by encrypting passwords through the hashing method used by Firebase authentication.
  - Securing the application from abuse of use and blocking users from the same IP address to register more than a predefined number per day and blocking a user from registering multiple times with the same email also has an effect on the consistency of the data.
- ❖ Performance Constraint: This application must:
  - Operate consistently under the pressure of multiple users' usage.
  - Have an acceptable response time which is set at 2-5 seconds.
- ❖ Error Management:

In case of errors recurring during its usage, the application should be able to:

  - Save this error log in the database so it would be fixed after.

- ❖ Maintenance Constraint:
  - The application code should be well commented on and easy to understand for reuse and update reasons.
  - The application code should be well modularized for a smoother maintenance process.
- ❖ Compatibility Constraint:
  - The application must be able to run on at least one of the operating systems which are iOS and Android.

- To allow optimal use of the application, it must use multiplatform available tools which can make the usage identic on both operating systems.

❖ Ergonomic Constraint:

- The user can manipulate the application without the need for prior knowledge since its user interfaces are very simplified. For example, to connect a device which is the same as searching for your Bluetooth speaker through Bluetooth, login is also generic same as most modern applications through social login, etc.
- The application should enable the user to achieve any task with the least number of clicks.

3.5. Use case Diagram

In the UML language, use case diagrams are used to model the behavior of a system or essential parts of a system and capture its requirements.

It defines the interaction between the system and an actor. Moreover, it specifies a series of actions taken by the system to produce an observable result that is relatable to a certain actor. The components of a use case are defined as the following:

- Actor: An actor is the representation of a role played by an external person, process, or another interacting system that interacts with the system.
- Use case: use-cases represent the functional relationships between the actors and the system.
- Association: it is used in this type of diagram to link actors and use cases by a relation which simply means “participates in”.
- Inclusion: The base use case explicitly incorporates another, so mandatory, at a location specified in its sequences.

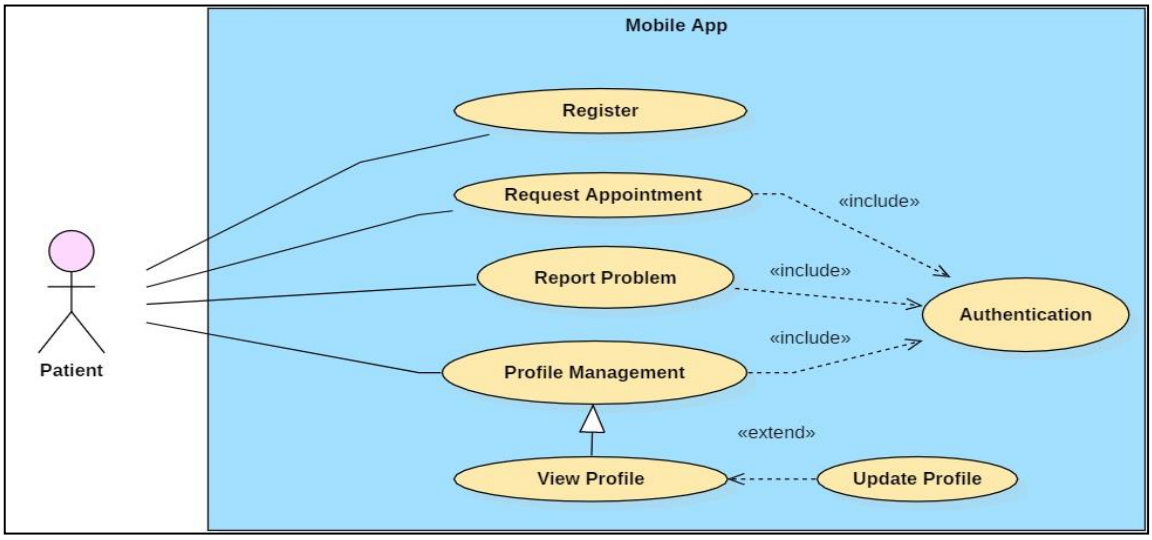

Figure 9 - Patient Use Case Diagram

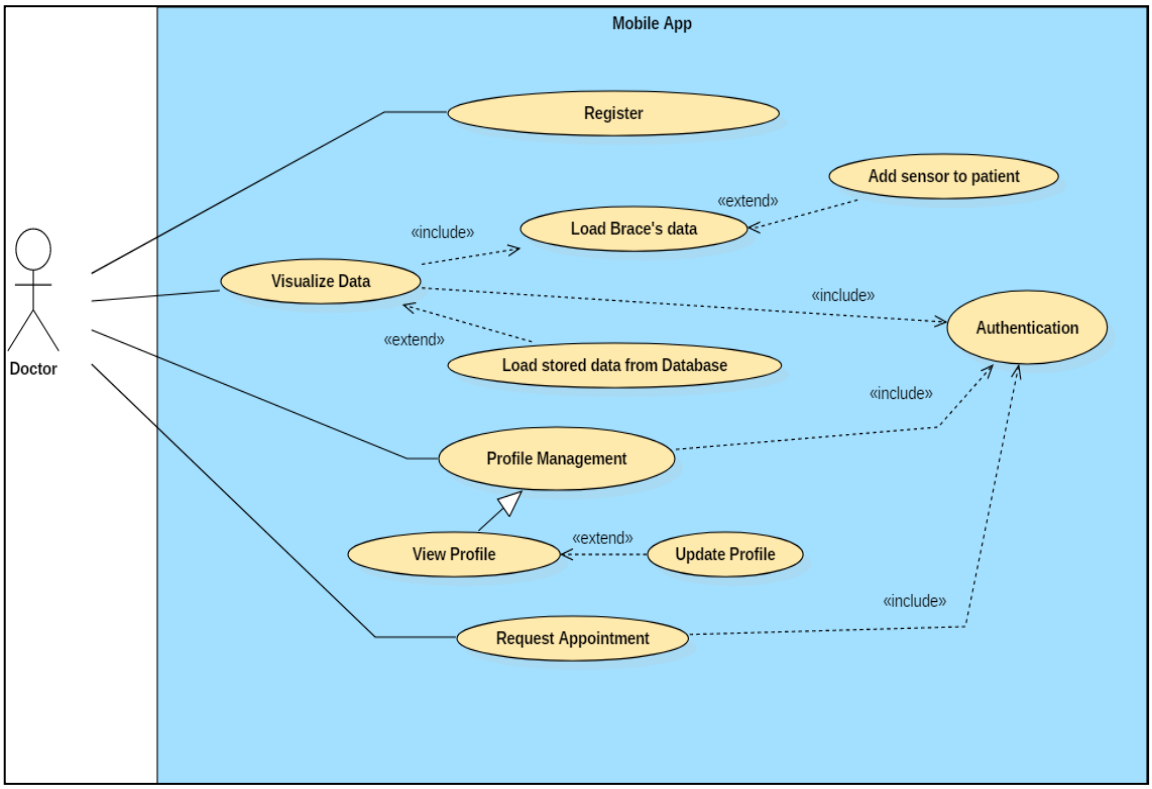

Figure 10 - Doctor Use Case Diagram

3.6. Planned workflow of the usage

After identifying the functional and non-functional requirements, the system specifications in this part will assist in defining the operational and performance criteria for a system using the block diagram illustrated in Figure 11. It will define how the system is intended to function and what that may involve through the various scenarios that the system should be capable of performing.

Going through this diagram, the doctor will start the application. Once it is open, he will have to log in through his credentials. In the case of the first use, he needs to register. After a successful login, he will be redirected to the home screen which is at first empty because he still didn't connect a device. Through the device's screen, he can connect to already paired devices or search for new devices. After finding and connecting to the needed device, he can go back to the home screen where he can visualize the real-time data coming from the device or select a date from the calendar to visualize that specified date's data and statistics.

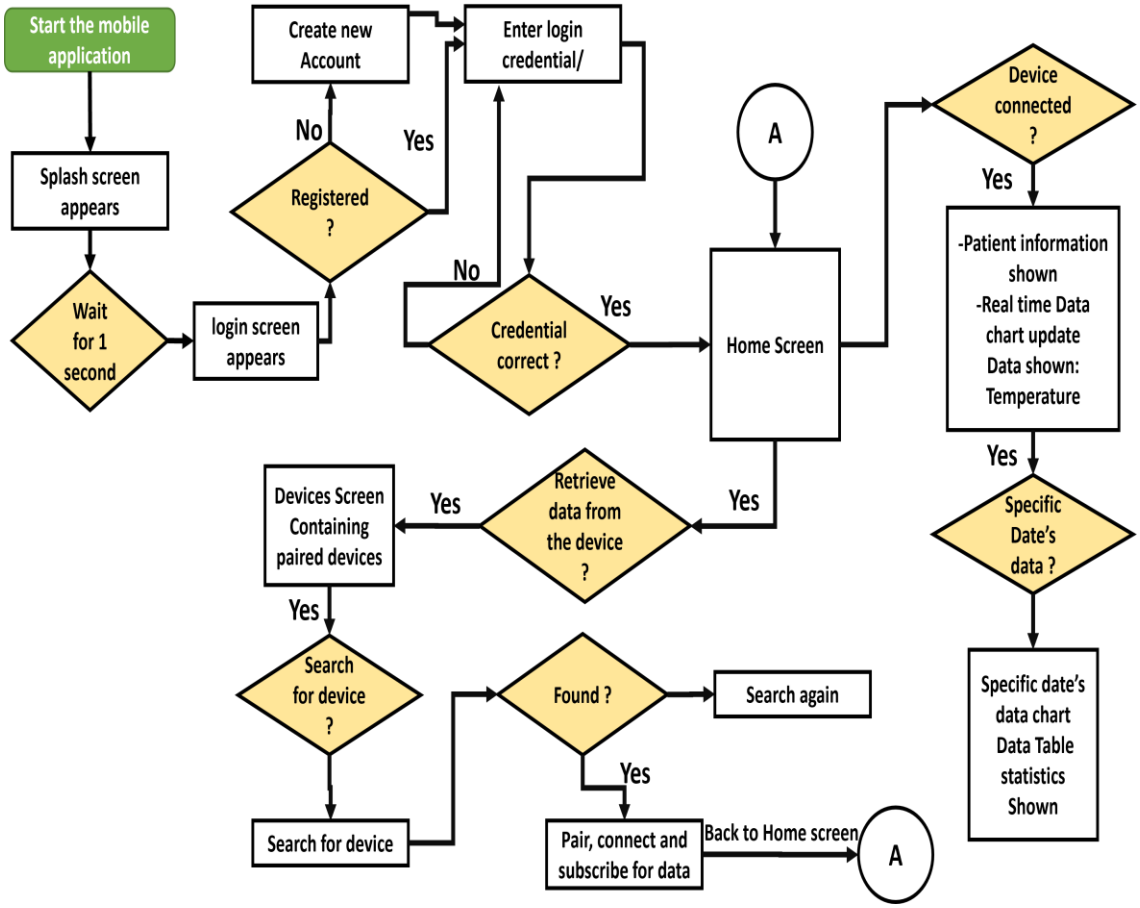

Figure 11 – Usage Flow Diagram

4. System Design

4.1. Purpose

In this section, I will put the spot on the definition of the system structure, its components, and its features.

4.2. Overview of the system design

The user who can be either the doctor or the patient can interact with the sensor-based braces through his phone using the cross-platform mobile application developed using Ionic Framework.

For the communication part, the application uses Bluetooth to establish a connection with the sensor inside the braces, and after that comes the retrieval of data and its display on the application UI.

In fact, to be able to implement such a feature which needs to be written in native code within the ionic framework, it needs to be implemented as a Cordova plugin.

The mentioned Cordova plugin uses native code, and it will be responsible for all interactions with the brace's sensor as well as the setup of the Bluetooth connection between the two ends. This plugin is called BluetoothLE which is an open-source plugin on Github providing all the needed functionalities to interact with devices using BLE.

In addition to connecting and transferring data wirelessly between the braces and the smartphone, the mobile application will be responsible for visualizing the data through a user-friendly interface utilizing tables and charts.

In terms of data storage, the application will be able to store the data retrieved from the braces into the Cloud Firestore Database which is provided by the Firebase Framework along multiple other features.

In Figure 12, I made a simplified diagram showing the different system components and the interactions between each other.

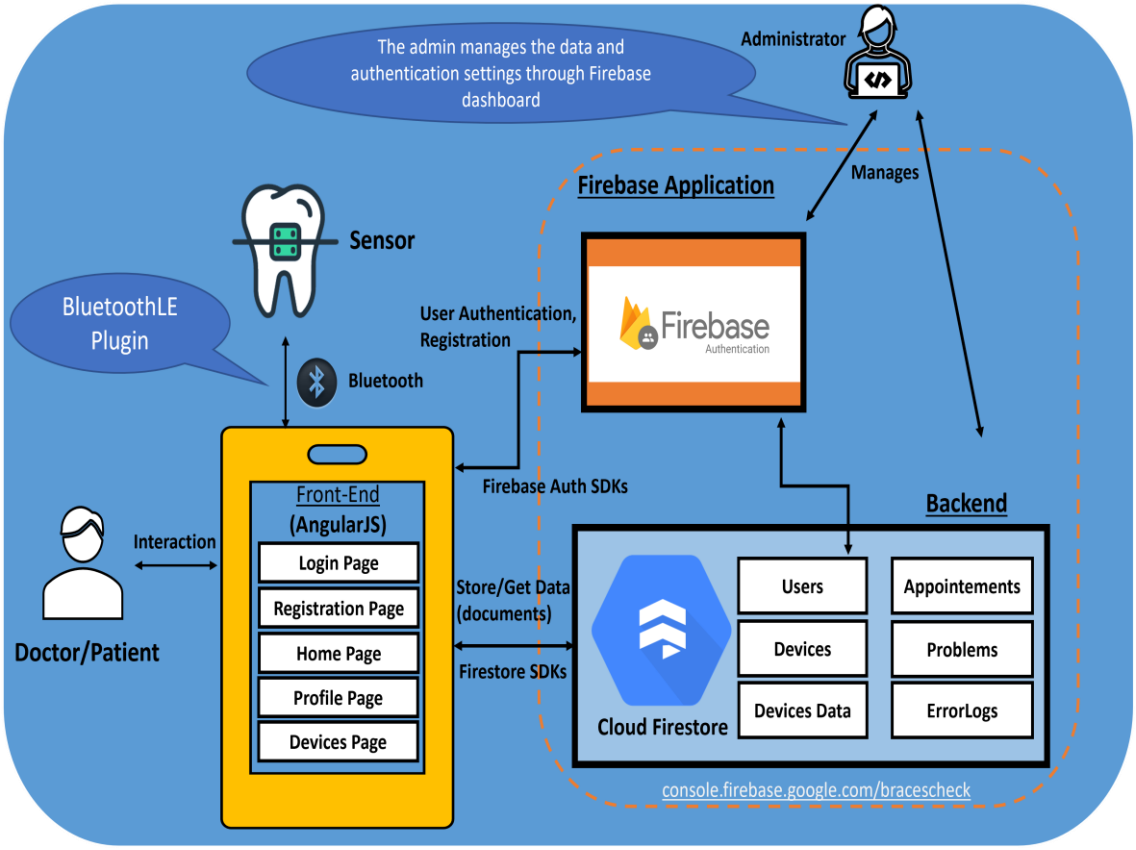

Figure 12 - System Design Specification

4.2.1. Firebase application usage

In the designed system, the management of the users is done by the administrator through the read-to-use dashboard of the Firebase platform. Through this dashboard, the administrator would be able to perform the needed management tasks (Create, update, and delete) on all the available models of the database including the user's model.

Of the features which were exploited also is the use of the Firebase authentication which enables to set the authentication's settings and SDKs to authenticate the user. Therefore, the authentication is resumed in the fact that the mobile just collects the user credentials and passes them to the Firebase Authentication SDK which will verify the credentials in the Firestore database and returns a response to the mobile application.

For the storing part, Firestore also provides SDKs to perform all the required tasks on the database. Since it is a NoSQL database, the process would be to pass the collected data related to certain models such as the device information in a Document format that contains the fields mapped to their values. This document will be then added to a Collection, which would be in our example "Devices", through a call to one of the ready to use methods which is for example "this.db.collection('devices').doc(this.appService.getProfile().id).collection('device').**add**(/* the key/values as parameter*/)". This applies to the other database transactions such as updating, deleting, etc.

4.2.2. Usage of Sensor-based braces

In this project, we used a microchip that stimulates the work of the sensor embedded inside the braces. This microchip has the same features such as wireless connectivity through the BLE technology, sends data continuously when connected to, etc. The mobile application is able to connect to the sensor by using the "BluetoothLE" plugin which was included in the application. This plugin allows the phone to interact with other BLE devices such as scanning for available devices, connecting to a device, subscribing to a service, etc.

In this project, the device's sensor basic information is stored in the "devices" model basic in the database. The primary identifier for a sensor is its physical address but it also has other information such as its name and the email of the patient related to it. Every device which was added already in the database won't be displayed on the scanning page, the doctor would be able directly to connect to it from the devices pages.

4.3. Design Specification

4.3.1. Back-End Specification

In this section, I will clarify the mobile Back-End structure, its components, how it interacts internally with each other, and how it runs as well. This phase is very crucial because the Backend part is where all the functions of the applications will be executed.

In order to clarify the mobile Back-End, I choose to make a conception of a UML sequence diagram which can give us a good idea about how the application can pair and establish a connection to a new sensor-based braces.

Description:

- Being already authenticated, the doctor wants a connect a patient's new device on the devices tab.
- The doctor will select the button add device on the upper screen and a popup will appear containing a list of the available devices on the Bluetooth grid after the scan is complete.
- The doctor will select the wanted device which can be identified by its name or its MAC address (physical address)
- the application will communicate with the braces sensor and a response will be sent back with the status of the request.
- If the device is no longer available, connected to another device, or got far from the Bluetooth range the application will display a toast stating that the device is not available.
- Else the application will pair, connect, and also ask the doctor for the patient's email through a pop-up so the device would be saved in the database with its owner.

- The doctor will have to fill out the email and confirm.
- The application will persist the device's data in the database.
- The application will show a toast message stating that the device is successfully connected.

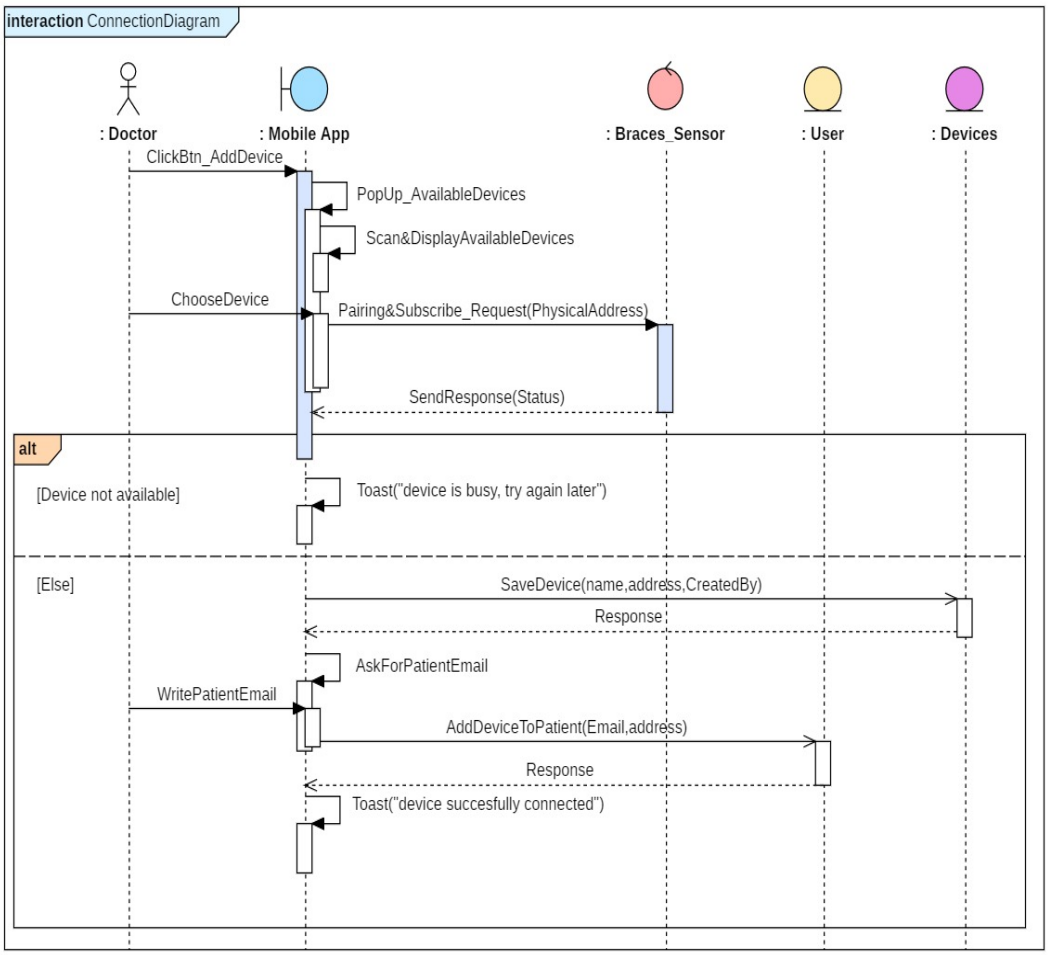

Figure 13 – Connecting to the device process Sequence Diagram

4.3.2. Front-end

To define the meaning of Front-end, it's everything on the user interface of an application or website that the user can interact with like the buttons, text boxes, etc. All these elements are built using HTML, CSS, and JavaScript.

The creation of a fluid user experience is one of the key aims of frontend development. Towards that goal in this section, I'm going to present my perception of a few prototypes for the user interface, easily manipulated and doesn't need an IT background to understand how it works.

In Figure 14, we can see the prototype of the home page which is the main targeted page of the application because it will contain the data visualization of the connected braces in the shape of charts, and also the doctor could see it in the shape of tables just

by swiping down in the home page. The data is mainly related to thermal data through time. On the doctor's home page, there are four sections:

- Patient and device information: This section contains the patient’s name, the brace’s name, the brace’s sensor MAC address, and the battery life percentage of the brace’s sensor.
- Realtime Temperature chart: This section contains a chart of the data received instantly from the brace’s sensor when it is connected.
- Old Temperature data: This section contains a calendar where the doctor can choose a certain date. By doing that, a chart and a table will appear under the calendar containing the temperature data recorded on that date with the exact time for every record. So even if the chart is condensed in case of a huge amount of data, the doctor can scroll in the data table.
- Statistics: This section will contain a chart which will also appear when choosing a certain date and it will show how many readings of data were done on that exact date.

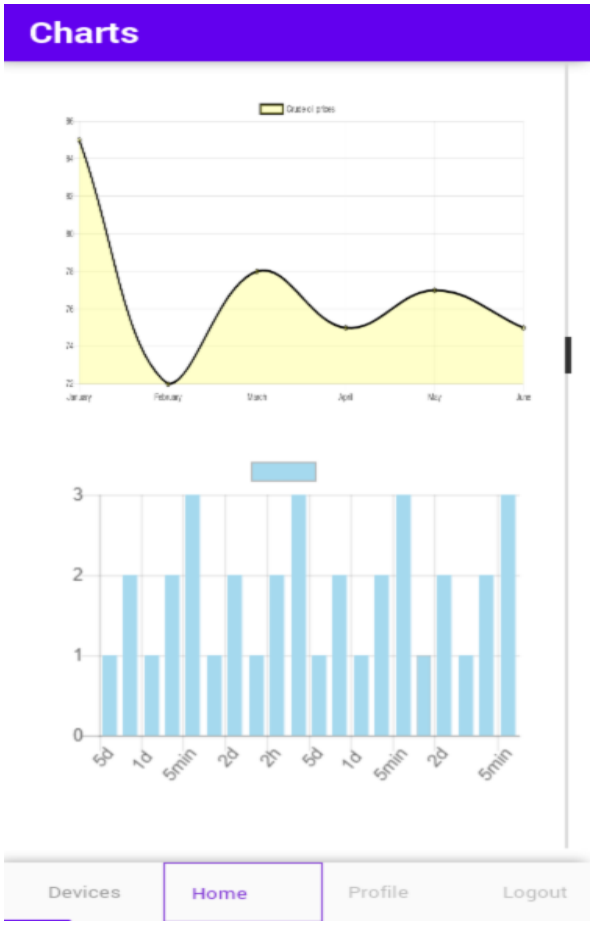

Figure 14 - Home page prototype

In Figure 15, we can see the prototype of the Devices page which is the first page used in the process of visualizing the data because it displays the already paired devices, it gives the possibility to add other sensor-based braces through the pair device page, and for every time a connection is established with a paired device it displays some useful information as the MAC address, the battery life, the state of the connection to the device, etc.

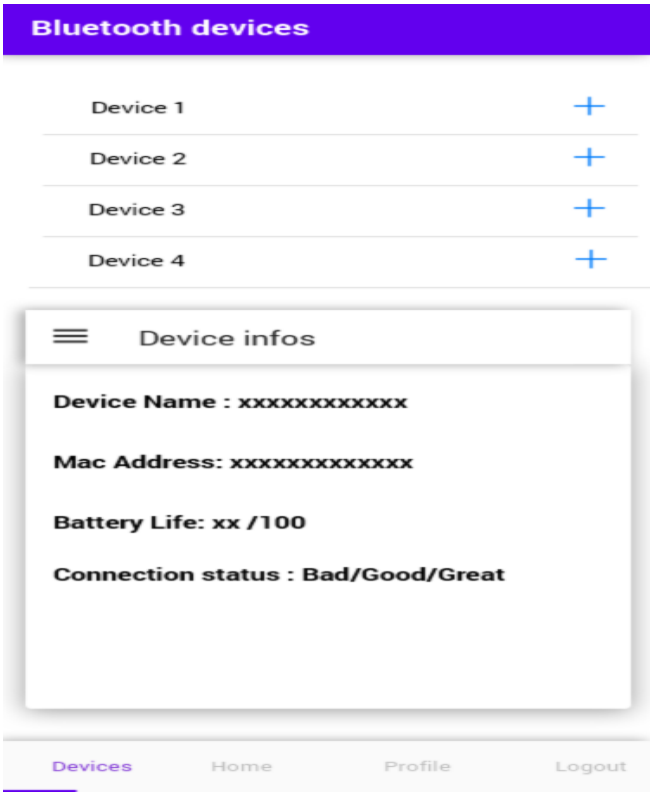

Figure 15 - Devices Page Prototype

4.4. System architecture

4.4.1. Data Storage

For my project, I used Firebase's Cloud Firestore. In fact, Cloud Firestore is a dynamic, scalable database from Firebase and Google Cloud for mobile, web, and server development.

For my specific project needs, I needed the following models:

- User: This is the main model which presents the user and his attributes such as his email, name, role (doctor or patient), date of birth, and sex.
- Devices: This model will contain the data related to devices paired through the usage of the application and which will be related to one patient.
- DevicesData: This model will contain as its name shows the data retrieved from the brace's sensor such as the measurements temperature, the movement motion per time, etc.
- Appointments: This model will contain the proposed appointment dates suggested by the patient in case of need.

- Problems: This model is also related to the patient, and it will contain his report messages in case of problems with the device his using.
- Error logs: This model will be responsible for persisting the log of errors occurring during the execution of the application so they can be traced back and fixed eventually.

To present the different models used in this project and the relations between them, I made a class diagram in the following figure:

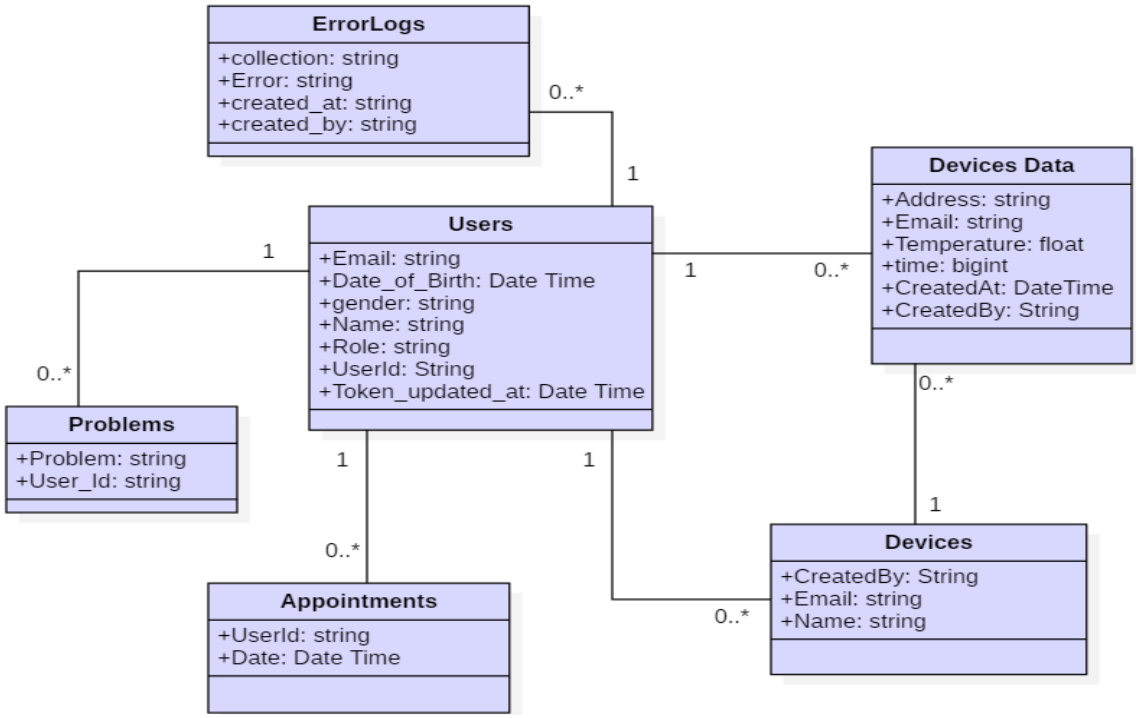

Figure 16 - Class Diagram

4.4.2. Data security rules

In the present era, data confidentiality and privacy are critical issues for users everywhere on the internet. It is getting increasingly difficult to keep data safe.

Many organizations and developers are implementing the Data Security process to protect system files, accounts, and databases by implementing a set of controls, techniques, and applications that define the relative importance of different datasets, and regulatory compliance requirements, and then implementing the measurements to secure the needed resources against dangers.

In this direction, I applied a certain security model consisting of 3 requirements:

- **Confidentiality:** The goal of this requirement is to secure the data from unwanted viewing and to ensure that only authorized users can have access to view it. To meet this requirement, I created an authentication mechanism that will occur before transferring data over the network. This was ensured in this

project through the usage of Firebase authentication which uses Jason Web Tokens and also a built-in hash method for passwords.

- **Availability:** The goal of this requirement is to ensure that the data is available whenever the need arises. This was ensured by using Firebase Framework for data persistence, which is rare a and highly available platform.
- **Integrity:** The goal of this requirement is to ensure and preserve data quality and integrity throughout its lifecycle. To that end, a constraint is in place to prevent unauthorized persons from altering data. More precisely, in this project, this was established by giving limited functionalities to every actor of the system, and even in this case, no functionality touches directly on the critical data of patients. As mentioned before, for example, the doctor has only the right to load previous data, and visualize the data, but he has no write permission on the data. Only the administrator has the possibility of deleting and updating the important data such as the devices' data.

4.4.3. Firebase Security Measurements:

Before working with Firebase services, I made sure it complies with major privacy and security standards. In fact, Cloud Firestore among the other Firebase services is certified under major privacy and security standards as you can see in the following figure taken from the official dedicated website of Firebase.

Service name	ISO 27001	ISO 27017	ISO 27018	SOC 1	SOC 2	SOC 3
Cloud Firestore	✓	✓	✓	✓	✓	✓

Figure 17 - Firestore Security Certifications

Apart from the certifications, Server-side rules that you create govern read and write access to specific pathways in your Firebase data tree and are enforced by Firebase security. Firebase security rules are JavaScript-like expressions that have access to the connection's credentials as well as the current view of the Firebase data tree, as well as any pending changes on write. In our project what we are looking for is securing the user and application data. This is where the Firebase Authentication comes in. Firebase Authentication is simply token generation: it takes verified, identifiable user data and securely transmits it to Firebase so that it cannot be spoofed. This validated credential information is subsequently made available in our security rules.

5. System implementation

5.1. Purpose

This chapter discusses the project's practical side, including the transformation of the product's plan and design. First and foremost, I will concentrate on the system development that I will consider. Then, I'll demonstrate how the code of the application works on both sides: android as well as the backend part.

5.2. System Development

After some consideration of the application's nature and the work environment, I decided to use the Waterfall approach in the application's development. The waterfall model illustrates a linear development process. It divides the development of software, applications, and web applications into five to seven phases. When one step is completed, the next one begins, with the results of the previous phases being incorporated into the new phase. Each phase has its own set of tasks and objectives, with all phases describing the software's life cycle until delivery.

The waterfall model is illustrated in Figure 18 along with its different steps:

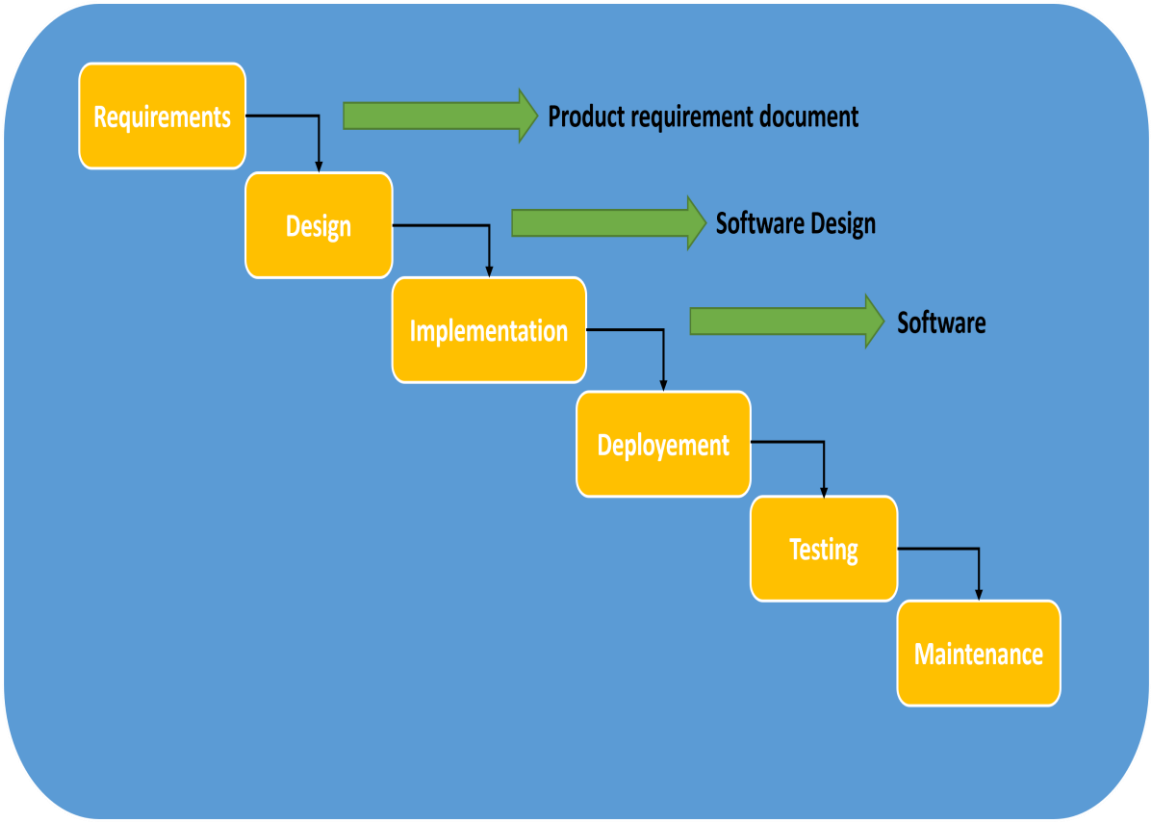

Figure 18 - Waterfall System development model

5.3. Characteristics of executing environments

- Windows 10 Operating System Laptop (Processor: Intel(R) Core (TM) i5-9300H , RAM: 24 GB, 64-bit operating system)
- The dental brace's Microchip
- Ionic version 5
- Cordova version 7
- Java 8 programming language
- Visual Studio code
- Android Studio
- Google Chrome for inspecting and Debugging
- OPPO CPH1911 running with Android 11

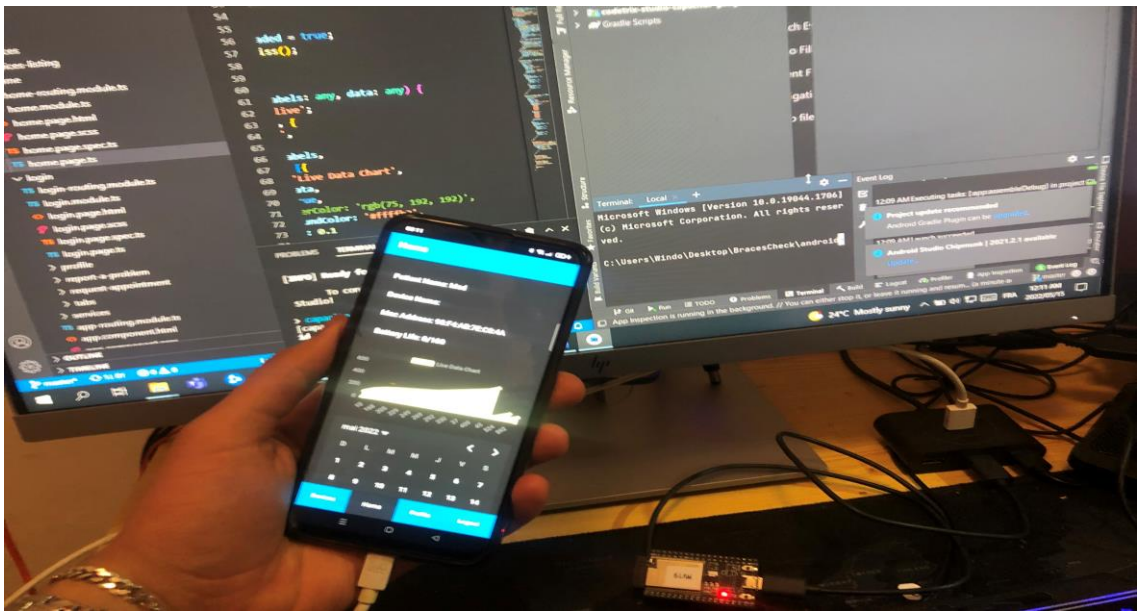

Figure 19 - Executing environnement

5.4. Ionic BLE Native Plugin

Before beginning to code the application, it is necessary to conduct a preliminary examination of the device's BLE technology. So, in that context, the services and characteristics of the device are the first things we need to know. Therefore, I have done some investigation through the internet, I used the well-known application BLE Scanner to scan the device and also by debugging the device through chrome inspect devices. So, to connect to the device we need the physical address of it like the one shown in the following figure:

Figure 20 - Device Inspection

Even though the found information is few, they are sufficient to carry out the development of the project or, at the very least, make what is linked to the Device clearer. I should be able to find more about the bracelet after certain characteristics are discovered.

After the connection is made, I could get more characteristics such as:

- The characteristic "2A19", this one contains the battery level of the device.
- The second characteristic I got was "4FAFC202-1FB5-459E-8FCC-C5C9C331914B", with this I can get the temperature data.

We know all there is to know about the device in order to retrieve all of the data needed.

The following figure shows the main structure of the services and the feature:

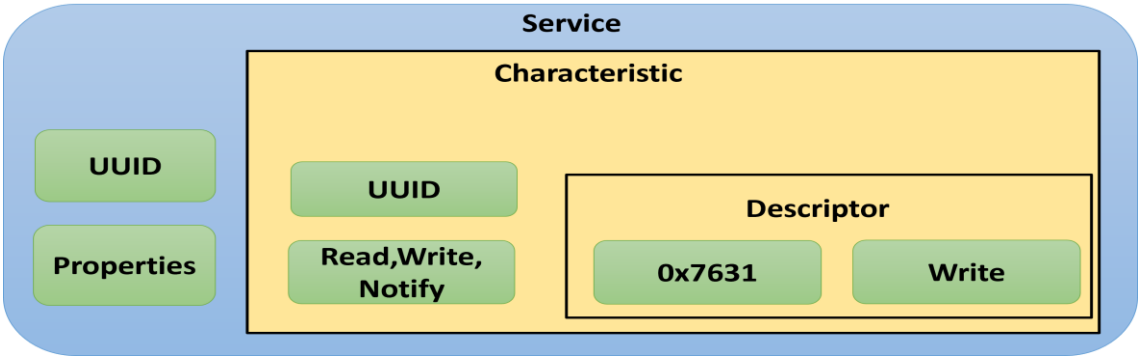

Figure 21 - Service Structure Example

In this section, I will demonstrate how the services, features, and so on are structured on the sensor of the dental braces, as well as illustrate that each feature has a descriptor.

Remark: For the device information service all the characteristics have only the Read property.

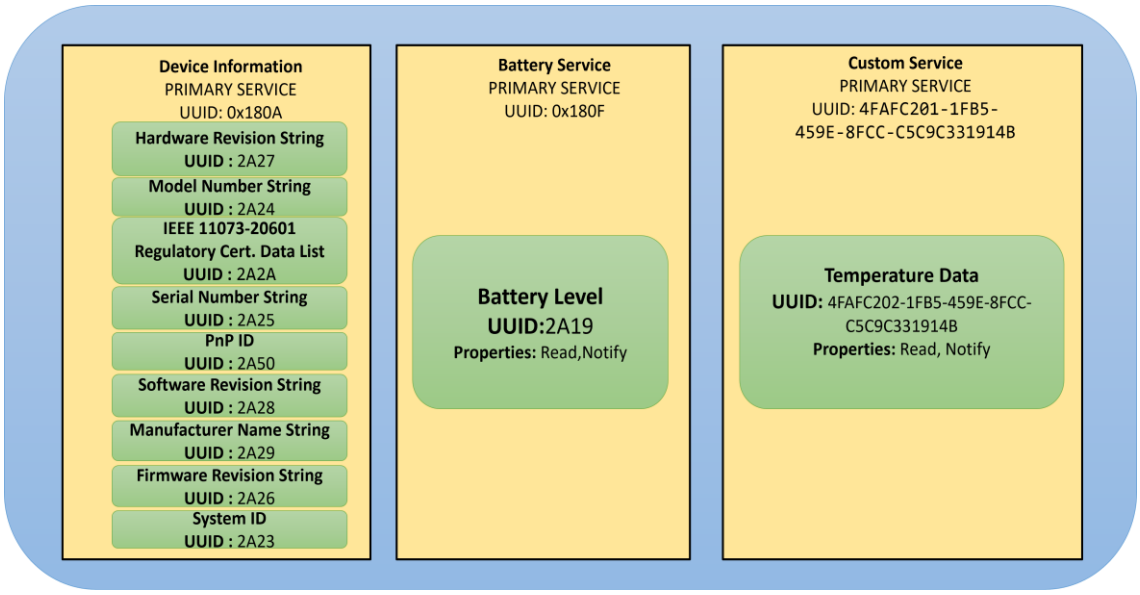

Figure 22 – Sensor Services Structure

The screenshot below shows the structure of services UUIDs which were used in my project alongside their characteristics.

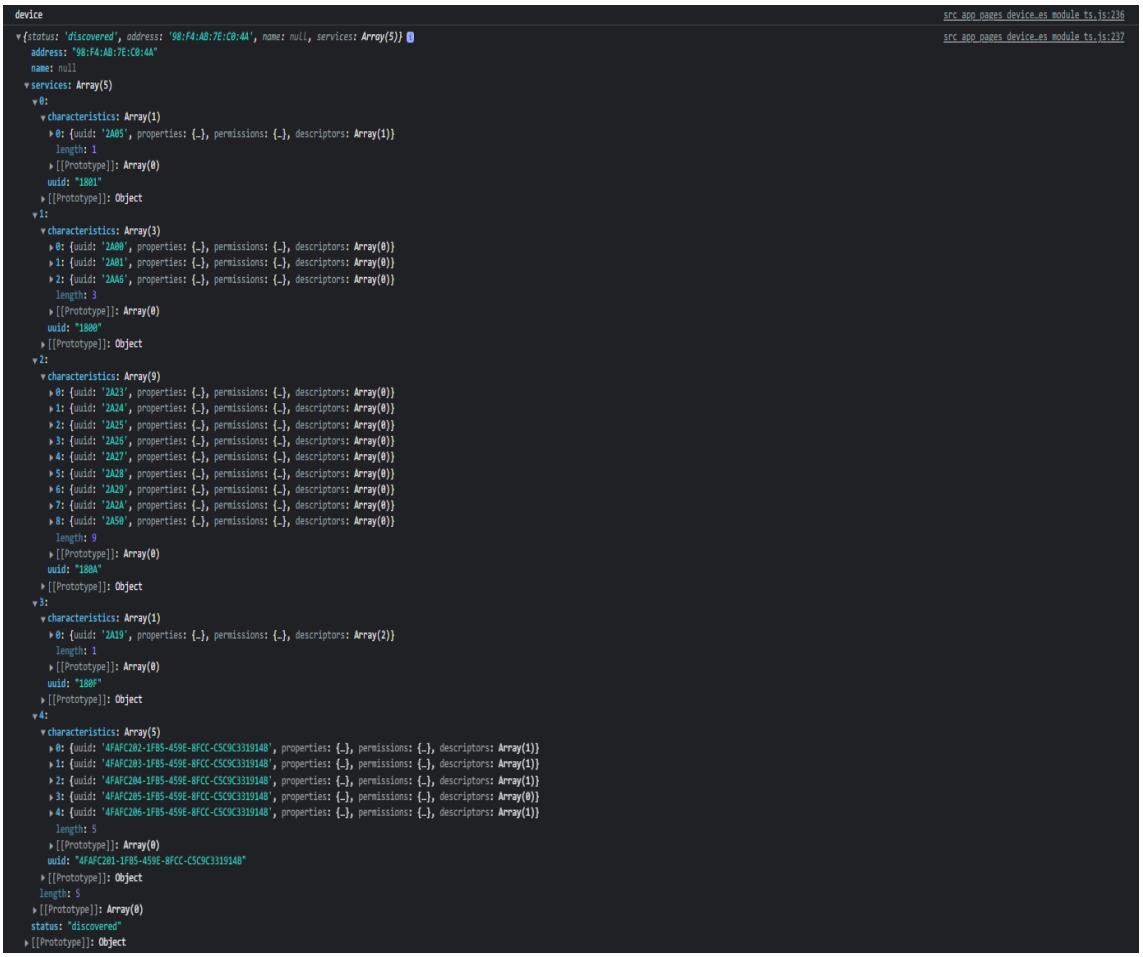

Figure 23 - Data Received

5.5. Database Implementation

In this section, I'll show you how the mobile application's local and backend database was implemented.

Before starting to build the application one of the main features that needed to be set was the backend service that will be used and also the service used for the local database side. Therefore, I chose Firebase due to multiple factors that I will shortly explain in this section.

From what we saw in Section **Erreur ! Source du renvoi introuvable.**, we can presume that Firebase Framework offers a lot of ready-to-use features within it which is favorable for managing my mobile application which requires:

Persistence of data: This is ensured through Cloud Firestore which enables cloud storage and synchronization, it’s easily accessible from mobile apps through native SDKs and it also supports Rest APIs.

User management and security: This is ensured by the Firebase Authentication which offers the possibility to authenticate the users securely through several ways like social login or email/password, and it also offers the possibility to set rules for the authenticating system as disabling the creation of multiple users with the same email.

In addition to these requirements, Firebase was chosen due also to its free usage plan for developers, limited human resources, and the size of the data used in this project. Therefore, supposing the application required more data storage space Firebase wouldn't be a good choice because as the data grows it becomes less effective in terms of querying it.

The graphical interface for managing the Firebase database is shown in the figure below:

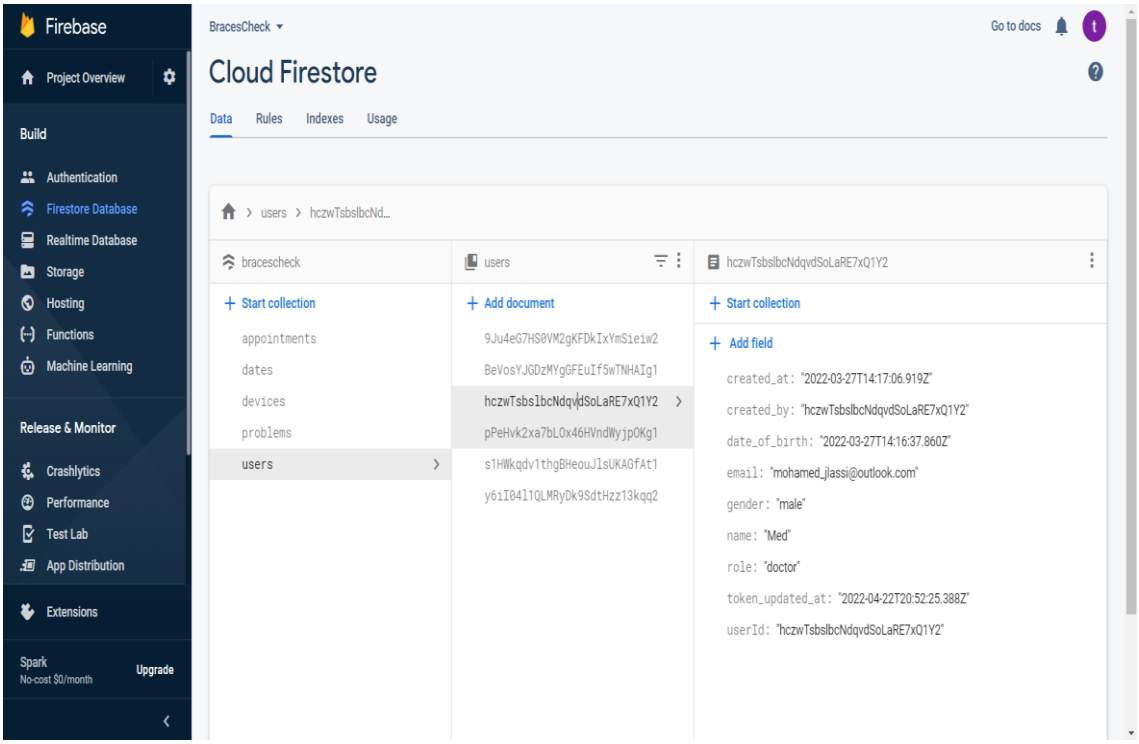

Figure 24 - FireStore Database Interface

As mentioned before the advantage of using Firestore is the ability to use other Firebase features which includes Firebase Authentication used also in our project.

Firebase Authentication is a robust token-based authentication mechanism. It facilitates the application's seamless integration with popular platforms such as Twitter, Facebook, and Google.

In Figure 25, we can see the interface of Firebase authentication which includes the user management, the sign-in providers, setup configuration for password resetting and email verification for example, and more useful options such as analytics that makes the management of user authentication smoother.

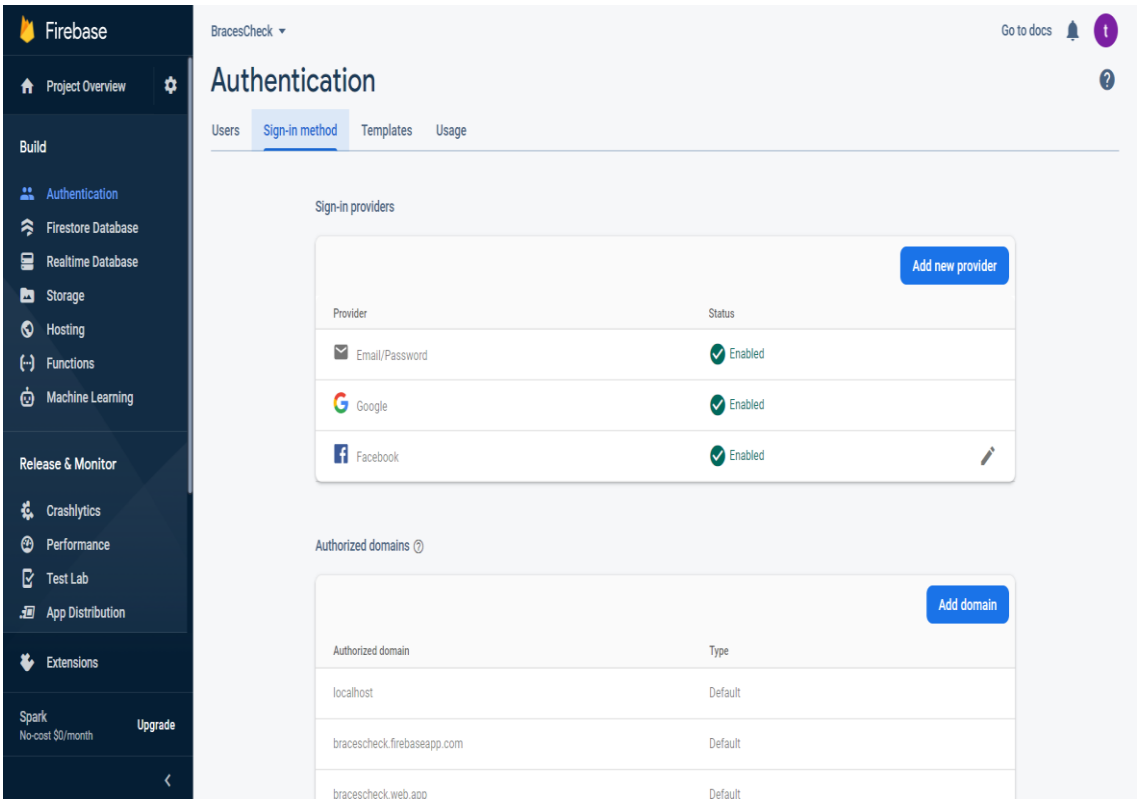

Figure 25 – Firbase Authentication interface

5.6. Mobile application

In this section, I will present the design that I used for my mobile application, as well as the various tabs that will be included.

When using the application for the first time, the user must create an account by filling out the registration form. This Registration form comprises all the user's personal information. As shown in the screenshot below, this information is Full Name, Date of Birth, Gender, Email, role (doctor or patient), and Password.

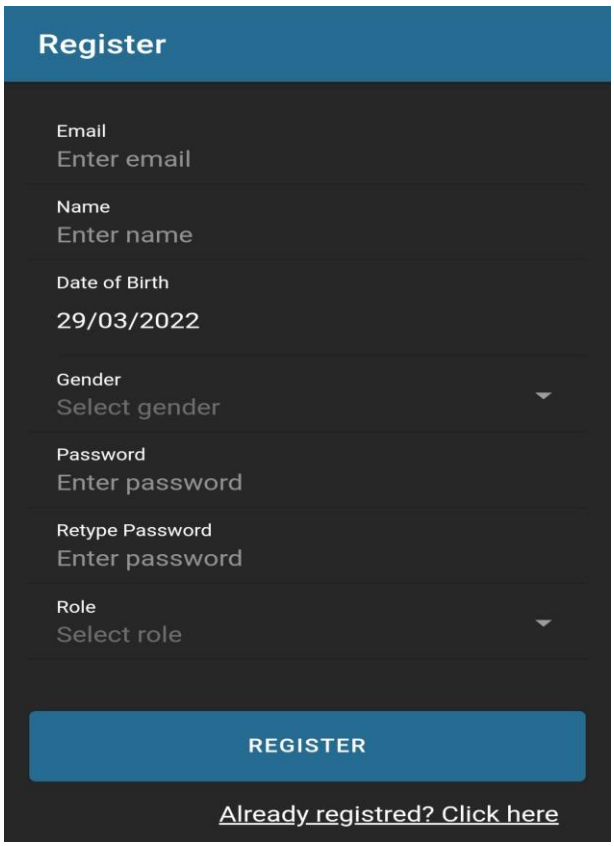

Figure 26 - Registration UI

After finishing the registration process, the user can log in to his account via email/password or social media options Facebook and Gmail. In our application, there are two types of users the doctor and the patient. So, when logging-in each user type will be redirected to different interfaces due to the different use cases for each role.

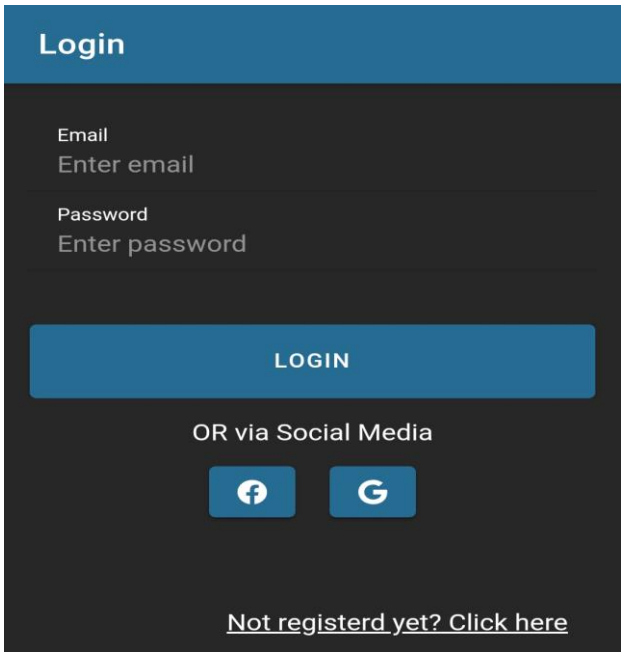

Figure 27 - Login Tab

For both roles, we will have 4 tabs after logging in. Two of these tabs are common which are the Profile tab and the Logout tab.

- **Profile Tab:** The profile tab is where the user can verify his personal information and alter or edit it to keep his information up to date.
- **Log Out Tab:** The user can log out of his account using this tab, and of course, all of his data would be already saved and stored in the database during the usage of the application.

On the other hand, we have two uncommon tabs between the two user roles and which I will explicitly clarify.

On the Doctor side of the application, we have:

- **Home Tab:** The home tab, is the main interface for the doctor where he is able to monitor the extracted data in both real-time and also previously stored data. The data concerned here is the temperature records during the usage of the braces. At the bottom of the home tab, we can find the statistics about the usage of the sensor.
- **Devices Tab:** In this tab, the doctor can see the already paired devices which are related to one patient each. In case of the need to pair a new device, the doctor can click on "Add new device" and a popup will show where he can scan for the available Bluetooth devices and pair the device needed.

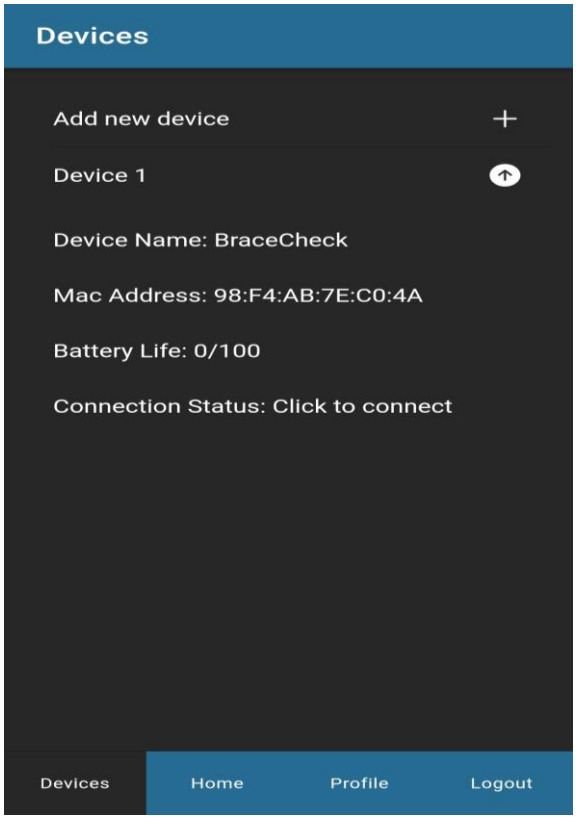

Figure 28 - Devices UI

On the side of the patient, we have:

- **Request appointment:** Using this tab, the patient can suggest a date to meet with the doctor in case of a problem or an urgent situation, and eventually in case of approval the patient can be informed via email later on.
- **Report a problem:** Via this tab, patients can report the problems they have with the braces and put a description for them so the problem can be verified and resolved. It's also a good way to help keep track in case of common problems affecting the patients.

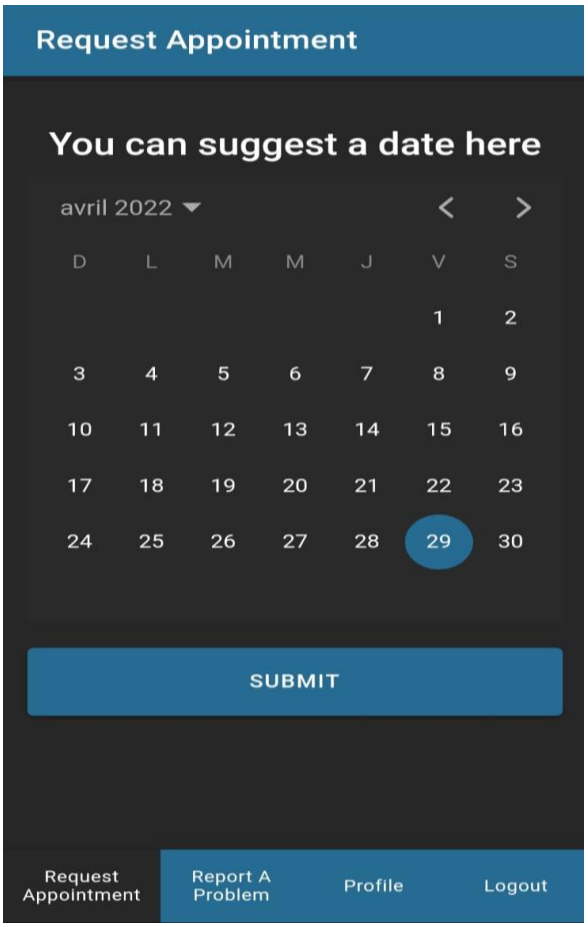

Figure 29 - Request Appointement UI

6. System testing

6.1. Purpose

For the application developed during this project to operate and give a consistent outcome, I need to provide appealing content and make sure that all of the functionalities operate smoothly and intuitively. As a result, whether deploying a mobile application or a desktop application, it is critical to conduct testing prior to deploying the developed application.

Although there is one key difficulty to testing a mobile app, which is adapting it to multiple devices, this process requires a lot more resources and effort for mobile apps than it does for other types of applications. Since this variety of phones was available in the early days of smartphones and mobile development, this adaptation was quite simple. However, nowadays, there are numerous different types of mobile devices, and each has its own set of features. Computers hardware differs from one computer to another and the same applies to smartphone models and tablet devices. In fact, the Random Access Memory (RAM), the processor (CPU), as well as the display and screen resolution, are the most important factors for an application.

On the other hand, there's the software difference that can be as important as the hardware in terms of its ability to help run the application smoothly.

8.1. Scope testing

At this point, we'll look at the framework from the standpoint of testing, and we'll look at a strategy for doing so. The method is to test the sub-framework units individually, compare their results to the intended outcomes, and then test the complete system together to ensure it is producing accurate results.

- Validation of the system:
  - Ensure the functionality of the system's sub-units.
  - Ensure the functionality of the combined system.
  - Ensure the data monitoring process.
- Verification of the system:
  - In the meantime, while making the system testing, I must ensure the verification of any system errors which should be eventually fixed.

6.2. Testing environment

In this section, I'll illustrate how this test is divided into two parts: functional testing and non-functional testing. Functional testing, as I mentioned earlier, is the first step in the testing process. During this phase, a unit test was established to verify and validate every feature of the code provided for the solution. When getting the results of each function, I may be required to compare them to those provided by the official application.

6.2.1. Functional Testing

In this phase, I will test each function of the system separately and provide the outcome of its task. The solution includes 6 functions, thus we will have 6 unique test cases along with their respective outcome and the obtained outcomes in case of a failed test scenario or a successful one, as shown in Table 4.

Test case	Test description	Successful test	Failed test
1-Authentication	This test case is divided into two parts: o Registration o Log in For the second part, we have 3 login options in our application: Login via the traditional email/password, Login via Facebook account, or Gmail account. **Expected outcome:** o User added/ user authenticated	Registration test: the user is added to the database and the user is redirected to the login page. Login test: the user is authenticated and redirected to the home page.	Registration test: the user is asked to verify the entered information in case of wrong typing or existing account. Login test: the user is asked to verify his credential and try again
2-User Management	This test case is divided only into 2 parts in our case because we don't have a module for deleting users. So, we only can test: o Adding a User: The addition of a user happens during the registration. o Updating a user: The update part is done through the update profile tab for every user, and he can update some information.	Adding a User: Same test outcome of registration in the previous test case (1). Updating a user: Same test outcome of Update Profile (test	Adding a User: Same test outcome of registration in the previous test case (1). Updating a user: Same test outcome of Update Profile (test case 5.2)

		case 5.2)	
3.1-Data Management (Load data from the dental braces)	Load data from the dental braces: The load data from the dental braces which are the temperature data is done through 2 phases which I tested: 1. Establishing connection to the sensor via BLE. 2. Extracting the data sent from the sensor through subscribing to the needed service inside it. **Expected outcome:** ○ Device paired and connected to the phone. ○ Device stored in the database if it's a new device ○ Temperature data received instantly	Establishing connection: device paired, connected to the phone. if the device is new, it will be saved in the database. Extracting the data: temperature data starts to be received instantly.	Establishing connection: if the Bluetooth is turned off, the device is far from the range limit, or the device is already connected to another device during the connection establishment it automatically fails. Extracting the data: requires the success of connection establishment
3.2-Data Management (Load stored data from the database)	This test aims to check if the data stored in the database is extracted properly to the application so it can be displayed on the home screen. **Expected outcome:** Data chart, data table, and stats related to a certain date are displayed on the home screen.	Load stored data: old data displayed in form of charts and a table.	N/A
3.3-Data Management (Save Data)	This test aims at ensuring that the data received from the device is stored in the database. **Expected outcome:** Data persisted in the model "devices Data" of the database	Data stored: we can check it also from the Firebase dashboard.	N/A
3.4- Data Management (Visualize	This test aims at ensuring that the charts, which are made for visualizing the data, contain the	Load stored data: live data displayed in	N/A

data)	right data and we can visualize it. **Expected outcome:** Data chart related to the live data sent from the sensor displayed on the home screen.	form of a chart.	
4- Add sensor to patient	This test case aims at checking that in case of a new brace's sensor has been added, this implies the doctor affecting it to a certain patient through filling his email. **Expected outcome:** The patient's email should be added to the device's information in the database.	The device is saved in the database and among its information is the patient's email address.	N/A
5.1-Profile Management (Profile Consultation)	○ Profile Consultation: Test if the user can visualize his profile data. **Expected outcome:** Profile page displayed containing the user information	Profile data displayed on the profile page	N/A
5.2-Profile Management (Profile update)	Profile Update: Test if the user can modify his data which implies the first test case, Profile Consultation, to be successful because it comes before updating the user profile. **Expected outcome:** Profile information is updated after submitting the changes in both the database and on the profile page.	Profile data changed on the profile page and persisted in the database.	If the typed information were empty or inadequate a toast of failure is displayed asking the user to verify the entered fields and try again.
6-Statistics Visualization	This test case is the same as the test case for the data visualization it aims at checking if the statistics are displayed in the chart with the correct data. **Expected outcome:** Statistics related to a certain date,	Statistics are displayed on a chart at the bottom of the home screen.	No chart is displayed in case the wrong date is chosen.

	chosen by the user, are displayed in a chart.		
7- Appointment suggestion	This test aims at suggesting an appointment date from a patient account and checking if the suggestion was recorded in the database. **Expected outcome:** Appointment suggestion persisted in the database.	The appointment suggestion is saved in the database and a toast is displayed confirming that.	In case of a wrong chosen date, a toast is displayed asking the patient to correct it and try again.
8-Problem Reporting	This test aims at filling a report from a patient account and checking if the report was recorded in the database. **Expected outcome:** Problem persisted in the database.	The problem is submitted and saved in the database with a confirmation Toast.	In case of an empty problem text, a toast is displayed asking the patient to fill the problem text.

Table 4 - Unit tests

6.2.2. Non-Functional Testing

Non-functional testing examines the system's non-functional features. It focuses on several aspects such as performance, usability, and dependability, among others. To put it in other words, it tests factors that are not examined during functional testing. Non-functional testing, on the other hand, is almost as necessary as functional testing. It also aids in improving the overall user experience.

We can find below the main points:

1- Security Constraint:

a) The application user passwords are encrypted through the hashing used in the Firebase Authentication service.

b) The application prevents users from creating different accounts through the same email address.

c) The application has a limit of 100/hour sign-ups from the same IP Address to prevent the project from abuse.

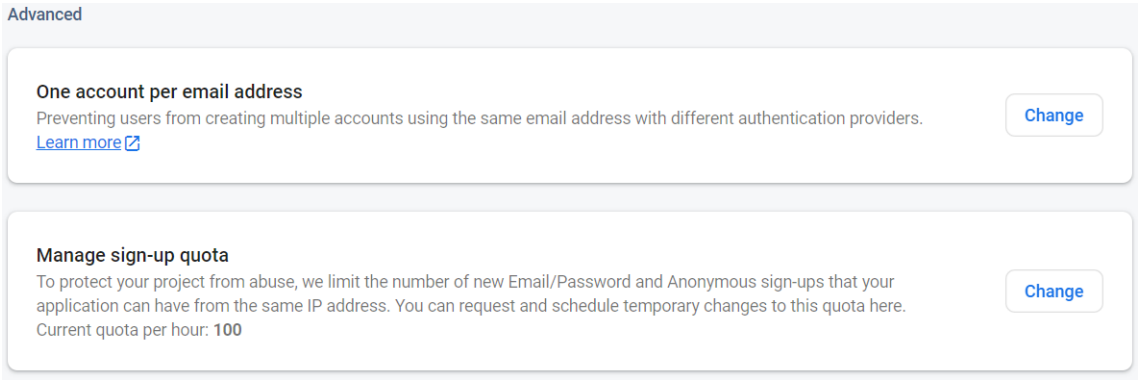

Figure 30 - Authentication Settings

2- Performance Constraint:

a) The application works consistently without errors by optimizing information relevance, robustness, and system response time which was approximatively under 5 seconds across each different tested functionality of the application.

b) The application enjoys an optimized access to data, and it is efficiently using resources without abuse.

3- Error Management:

The application is able to log the errors generated during its usage and saves them in the database with a description and the exact time. (Tested and the errors can be found in the database)

4- Maintenance Constraint:

The code is clear and well commented. Also, the code is well divided into clear modules. So, any future updates or improvements to the modules or the structure are possible.

5- Compatibility Constraint:

The application runs properly actually on android.

6- Ergonomic Constraint:

The user of the application doesn’t need to have a particular knowledge or training to be able to manipulate the application thanks to its simple user interface and the simplicity of completing any task through the user interface.

6.2.3. System testing

In this phase, we should inspect and check that the entire system and all of its components are functioning properly. The features we should expect to see are as follows: Logging-in, Establishing the connection to the device and eventually visualizing the data.

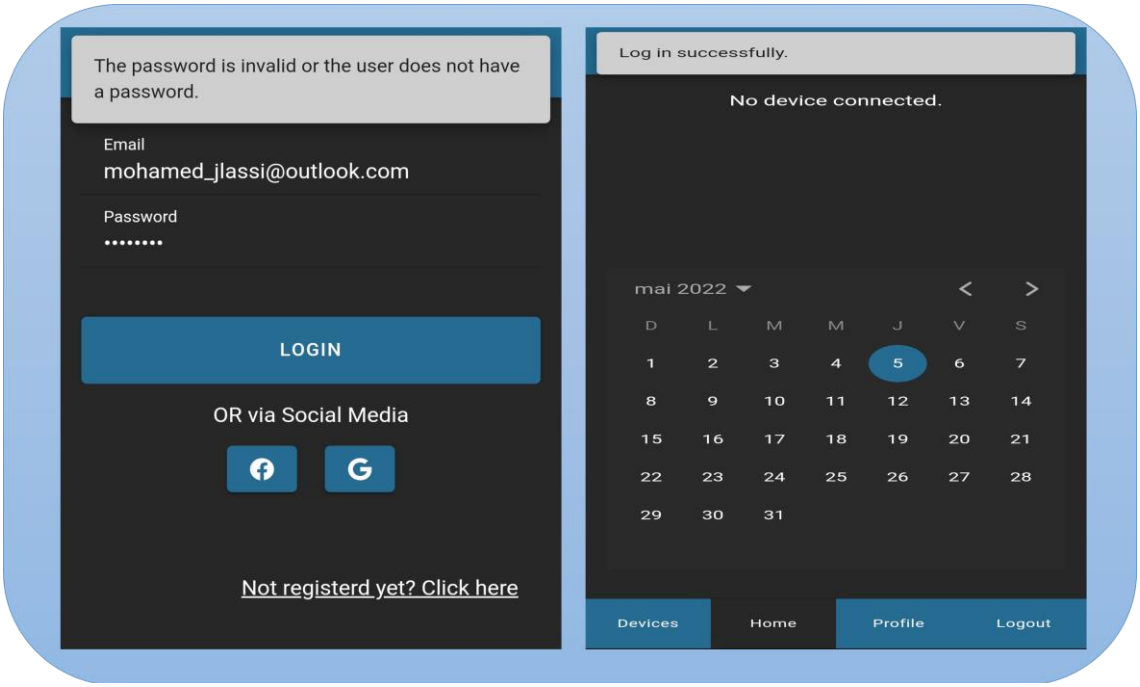

Figure 31 - System testing: Login

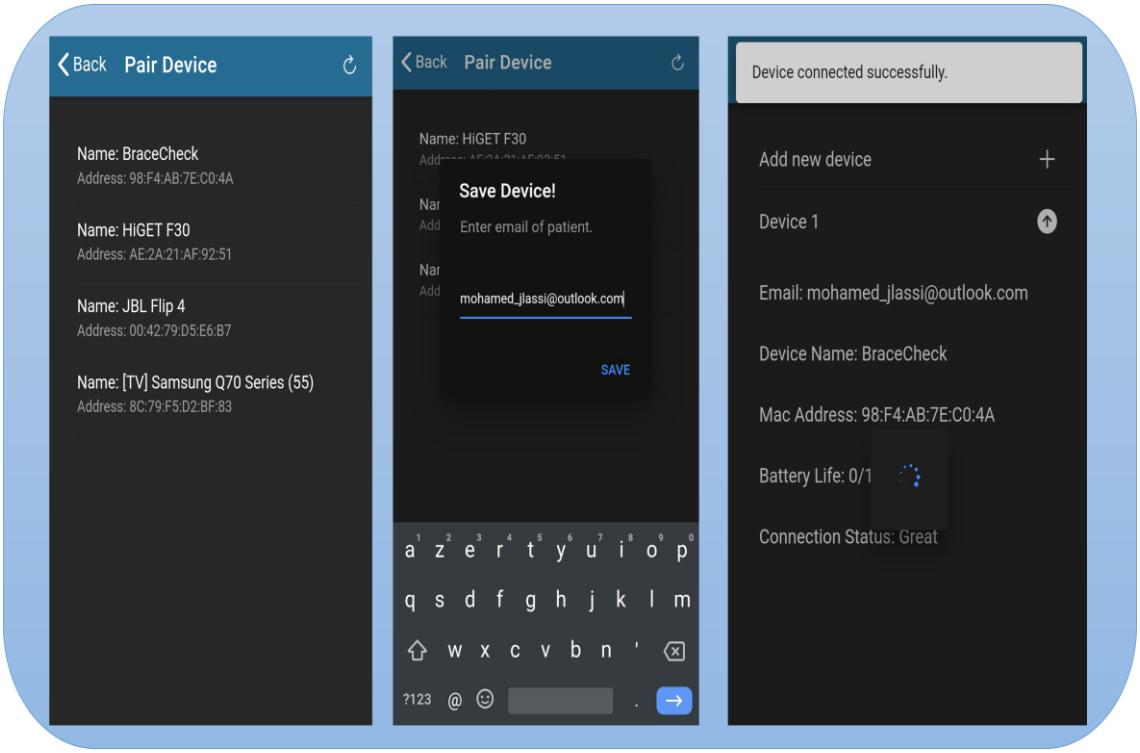

Figure 32 - System testing: Pairing and Connecting to device

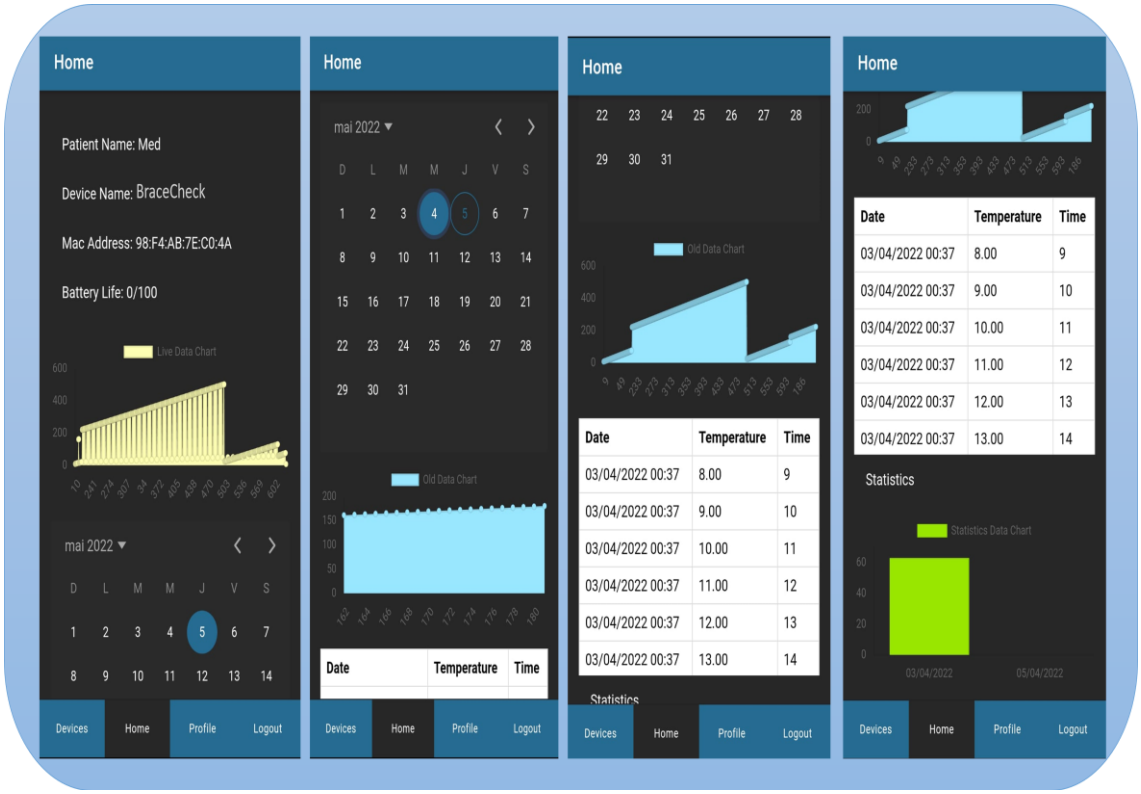

Figure 33 - System testing: Visualize data

6.3. Backend Testing

In order to carry out a test on the backend database, I performed some tasks that would affect the database data, which should result in queries, and ensure the data persistence.

As we can see after executing the test in the following figure, the temperature data is being persisted in the database along with other data related to the patient and the device where the data is coming from:

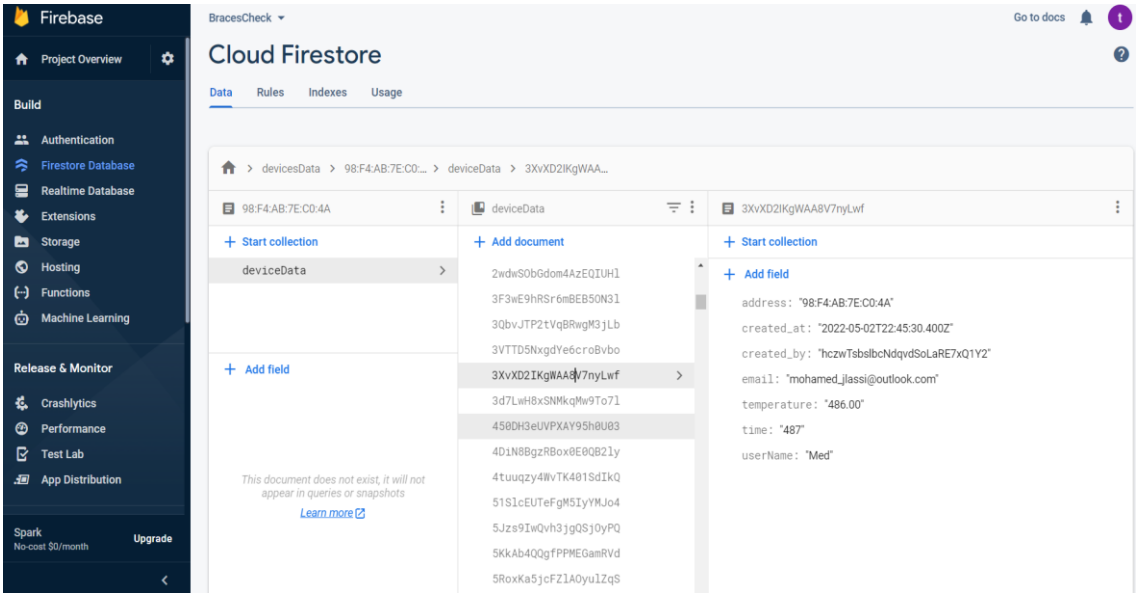

Figure 34 - Data persistence Test

6.4. Performance testing

To measure the application performance, I used the Apptim testing application for mobile apps to test all the functionalities available on the application during a test session. After I finished the testing session which lasted 1minute:11seconds, we can see in the following figure the result of the test session.

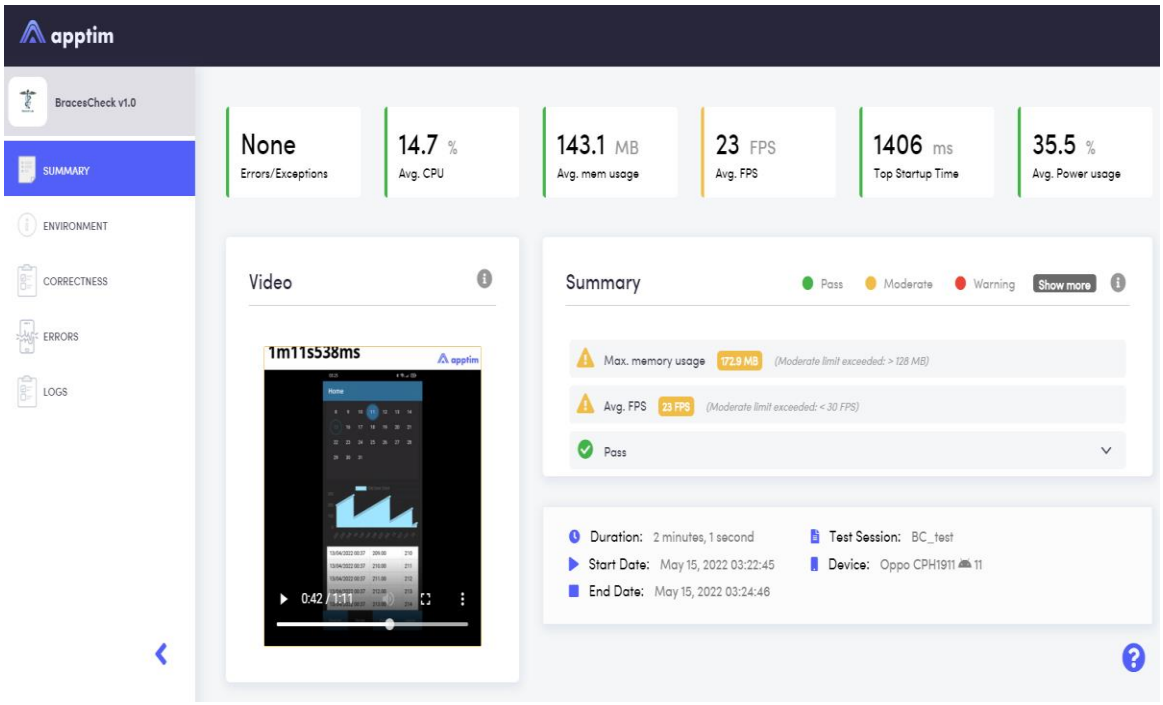

Figure 35 – Apptim Performance test result

6.5. Future Work

Finally, this project has succeeded in all the phases from development to testing. But it can be the foundation for more improvements and addition in terms of new features and optimization in the performance. Given the simplicity of the code and the appliance of modularity during the development, the task of maintenance and making improvements will not be complicated.

So, on this matter I wanted to suggest some propositions for the future work on this project which are:

- Build more versions of the application such as an iOS version and desktop version.
- Add a notification system for reminders about appointments, low battery percentage, etc.
- Add more useful statistics related to the patient such as braces usage routine.
- Add More features to the patient part of the application where he can receive notifications for example when his suggestion for a meeting was approved or not.
- For this thesis work, we didn't have a large set of data that's why Firebase was convenient. Therefore, in the case of larger datasets, Firebase would have some difficulties querying and the solution would be an additional database that could be used to handle the persistence of the data such as AWS RDS.

Remark: The GitHub repository for the BracesCheck application is:

github.com/JlassiMed/BracesCheck

Conclusion

As a result of this thesis work research, which is dedicated to obtaining the master's degree in Computer Science Engineering at the Biotech Research Center at Óbuda University, I was able to design and develop a mobile application that aims at monitoring the dental braces treatment through connecting to the braces embedded sensor through the BLE technology, extract the data, and visualize it in simple form as it will be the base for the dentist to make decisions. This data is also stored online on the Cloud Firestore Database to ensure the persistence of the security and high availability of the data.

The literature study and research that I conducted to gather all of the material required to gain a thorough understanding of the traditional dental braces, how it works, and how monitoring them can help dentists make more efficient decisions and reduce the time of the treatment for the patients and the critical need for such a system enabled me to develop this solution. On the other side, developing this project was a very unique experience as I got in touch with new technologies such as Ionic, BLE, AngularJs, and others which gave me the possibility to be able to reach a good level of knowledge in these technologies and that what led me to accomplish the needed functionalities of the application alongside the expected results. The system went through several phases of testing and validation of its functionalities with a detailed description for each test and the outcome of it alongside some screenshots.

The system developed during this thesis work was a sort of combination of two crucial subsystems. The first was the dental braces containing an embedded sensor to get the data and this sensor can communicate certain data about the braces through the BLE technology. The second subsystem was the mobile application which enables the doctor to connect to the brace's sensor, extract the data wirelessly, display it on the phone, and in the background store it in the Cloud Firestore Database so it can be loaded, in case of need later on. However, most of the functionalities of the application are dedicated to the doctor, I developed a part for the patient where he can suggest appointments and report his technical problems with the braces. This patient part can be the base for other improvements as I mentioned before in the Future work section.

Acknowledgment

Firstly, I would like to thank ALLAH, for helping me finish this thesis work and for supporting me with patience and belief.

Then, I would like to mention a special thanks to my supervisor Dr. Miklós Kozlovszky, who assisted me throughout the whole thesis work period and provided me with all the needed guidance and motivation for the completion of it.

Also, I would like especially to thank my mother for her continuous encouragement and belief in me, also my father, all my family, my friends who are living in Hungary or back at home, and any person who contributed to this achievement directly or indirectly.

I want likewise to thank my dearest university, Óbuda University, John von Neumann Faculty of Informatics for the quality of the provided curriculum and the multicultural environment offered for the students.

Finally, I would like to thank the Tempus Public Foundation and the Tunisian Ministry of Higher Education for the opportunity they gave me to be a part of the Stipendium Hungaricum scholarship to pursue my studies in Hungary.

List of Figures

List of Tables

References

[1] M. J. Baheti and N. Toshniwal, “Orthodontic apps at fingertips,” *Prog Orthod.*, vol. 15, no. 1, p. 36, Dec. 2014, doi: 10.1186/s40510-014-0036-y.

[2] “Statista - The Statistics Portal,” *Statista*. https://www.statista.com/ (accessed May 04, 2022).

[3] “An empirical study of wearable technology acceptance in healthecare.pdf - The current issue and full text archive of this journal is available on | Course Hero.” https://www.coursehero.com/file/64145176/An-empirical-study-of-wearable-technology-acceptance-in-healthecarepdf/ (accessed May 04, 2022).

[4] H. Lewy, “Wearable technologies – future challenges for implementation in healthcare services,” *Healthcare Technology Letters*, vol. 2, no. 1, pp. 2–5, 2015, doi: 10.1049/htl.2014.0104.

[5] I. V. B. Team, “Wearables: interesting for the dental world?” https://www.blog.ivoclar.com/dentist/en/wearables-interesting-for-the-dental-world (accessed May 04, 2022).

[6] S. Caruso, S. Caruso, M. Pellegrino, R. Skafi, A. Nota, and S. Tecco, “A Knowledge-Based Algorithm for Automatic Monitoring of Orthodontic Treatment: The Dental Monitoring System. Two Cases,” *Sensors (Basel)*, vol. 21, no. 5, p. 1856, Mar. 2021, doi: 10.3390/s21051856.

[7] “How Dental Apps Improve Oral Health [Infographic],” *DentaVox Blog | Dental Stats & Paid Surveys Info*, Mar. 11, 2020. https://dentavox.dentacoin.com/blog/how-dental-apps-improve-oral-health-infographic/ (accessed May 04, 2022).

[8] A. Bianco *et al.*, “COVID-19 and Orthodontics: An Approach for Monitoring Patients at Home,” *The Open Dentistry Journal*, vol. 15, no. 1, Mar. 2021, doi: 10.2174/1874210602115010087.

[9] “The History of Orthodontics: From Ancient Braces to Invisalign,” *Central Coast Orthodontics*, Jul. 15, 2021. https://centralcoastorthodontics.com.au/the-history-of-orthodontics-from-ancient-braces-to-invisalign/ (accessed May 03, 2022).

[10] A. A. Atanasova and M. A. Varneva, “Braces- Another Way of Treatment of Dentofacial and Maxillofacial Deformities,” vol. 5, no. 3, p. 6, 2018.

[11] T. Rahman and J. R. Bowen, “(54) BRACE COMPLIANCE MONITOR,” p. 6, 2005.

[12] "Dental Monitoring," *Ilam Orthodontics*. https://www.iortho.co.nz/dental-monitoring/ (accessed May 13, 2022).

[13] C. A. Brierley, P. E. Benson, and J. Sandler, "How accurate are TheraMon® microsensors at measuring intraoral wear-time? Recorded vs. actual wear times in five volunteers," *Journal of Orthodontics*, vol. 44, no. 4, pp. 241–248, Oct. 2017, doi: 10.1080/14653125.2017.1365220.

[14] "Connect to Tech or Dentist Environment," p. 9, 2019.

[15] A. Lo Giudice *et al.*, "Teleorthodontics: Where Are We Going? From Skepticism to the Clinical Applications of a New Medical Communication and Management System," *Int J Dent*, vol. 2022, p. 7301576, Feb. 2022, doi: 10.1155/2022/7301576.

[16] "Firebase vs. Firestore | What are the differences?," Aug. 20, 2020. https://blog.back4app.com/firebase-vs-firestore/ (accessed Apr. 29, 2022).

[17] "Firebase Authentication | Firebase Documentation," *Firebase*. https://firebase.google.com/docs/auth (accessed May 02, 2022).

[18] "Cloud Functions for Firebase | Firebase Documentation," *Firebase*. https://firebase.google.com/docs/functions (accessed May 02, 2022).

[19] "Cloud Storage for Firebase | Firebase Documentation," *Firebase*. https://firebase.google.com/docs/storage (accessed May 02, 2022).

[20] R. Heydon, *Bluetooth low energy: the developer's handbook*. Upper Saddle River, NJ: Prentice Hall, 2012.